



Telecom & IT Solutions

CMM

- Nível 2 -



Versão 1.2

11/10/2001

Tradução

José Marcos Gonçalves

André Villas Boas

Índice

NOTA DOS TRADUTORES.....	I
DAS RESPONSABILIDADES	II
DISCLAIMER.....	IV
1. GESTÃO DE REQUISITOS	1
2. PLANEJAMENTO DE PROJETO DE SOFTWARE	9
3. ACOMPANHAMENTO E SUPERVISÃO DE PROJETO DE SOFTWARE	25
4. GESTÃO DE SUBCONTRATAÇÃO DE SOFTWARE.....	38
5. GARANTIA DA QUALIDADE DE SOFTWARE.....	52
6. GESTÃO DE CONFIGURAÇÃO	62

Nota dos Tradutores

Este trabalho tem como objetivo único facilitar a divulgação de um modelo de melhoria de processos que vem sendo muito utilizado e tem trazido resultados positivos àqueles que o estão usando.

O interesse no assunto é fruto de um trabalho que vem sendo realizado na Fundação CPqD, na área da qualidade, com a avaliação de vários modelos de melhoria de processos de software e posterior adoção do CMM. Após muitos estudos, várias palestras, adaptação do modelo para o CPqD e execução de mini-avaliações, além de participação em uma avaliação oficial do SEI, foi observado que um dos principais problemas de divulgação das práticas do modelo residia no fato do mesmo estar escrito em inglês. Diante disto, decidiu-se traduzi-lo e torná-lo público, esperando com isto facilitar o acesso de mais pessoas ao assunto.

Esta não é uma tradução oficial do documento do SEI e qualquer uso formal do CMM deve ser feito com base na documentação oficial do SEI, a qual pode ser obtida no endereço: <http://www.sei.cmu.edu>.

A Fundação CPqD não se responsabiliza pelo uso do material aqui contido, já que o mesmo tem caráter apenas informativo.

Todo e qualquer comentário, bem como o interesse em receber versões aprimoradas deste texto, podem ser enviados para os tradutores que, desde já, agradecem a atenção.

Um abraço e bom trabalho,

André Villas-Boas (villas@cpqd.com.br)
José Marcos Gonçalves (jmarcos@cpqd.com.br)

Campinas, outubro de 2001.

Outras traduções relacionadas:

- TR 24 – O Modelo CMM (disponível)
- TR 25 – Interpretando o CMM (em elaboração)
- TR 25 – Nível 3 (em elaboração).

Das Responsabilidades

Uma permissão especial para reproduzir material sobre o Capability Maturity Model® (CMM®), que representa direitos autorais da Universidade Carnegie Mellon é concedida pela Universidade Carnegie Mellon.

® Capability Maturity Model é uma marca registrada da Universidade Carnegie Mellon.

® CMM está registrado no Escritório de Marcas e Patentes dos Estados Unidos da América.

DA GARANTIA. O MATERIAL DA UNIVERSIDADE CARNEGIE MELLON E DO INSTITUTO DE ENGENHARIA DE SOFTWARE É FORNECIDO EM BASES “AS IS”. A UNIVERSIDADE CARNEGIE MELLON NÃO FORNECE QUALQUER GARANTIA SOB NENHUMA ESPÉCIE, SE EXPRESSA OU IMPLÍCITA, A QUALQUER TEMA INCLUSO, MAS NÃO LIMITADA A, GARANTIA DE ADEQUAÇÃO PARA O PROPÓSITO DESTE OU MERCADOLÓGICO, EXCLUSIVIDADE OU RESULTADOS OBTIDOS DO USO DO MATERIAL AQUI CONTIDO. A UNIVERSIDADE CARNEGIE MELLON NÃO OFERECE QUALQUER GARANTIA, SOB QUALQUER ESPÉCIE, COM RESPEITO A ESTAR AS PATENTES, MARCAS OU DIREITOS AUTORAIS LIVRES DE QUALQUER INFRAÇÃO.

Da Isenção de Responsabilidade. A Fundação CPqD assegura à Universidade Carnegie Mellon a isenção de todas as responsabilidades demandadas, danos, despesas e perdas ocorridas de uso negligente ou uso indevido deste, ou conduta indevida pela Fundação CPqD considerando a Propriedade Intelectual da Universidade Carnegie Mellon. Essa cláusula deve permanecer após o término desse Acordo.

Do Foro. Esse Acordo deve ser governado pelas leis da Comunidade da Pennsylvania. Qualquer disputa ou reclamação ocorrida disso ou em relação a esse Acordo terá arbitragem estabelecida em Pittsburgh, Pennsylvania (EUA), de acordo com as regras da Associação Americana de Arbitragem e com julgamento prestado por árbitro(s) em qualquer corte de sua(s) jurisdição.

Se o material da Universidade Carnegie Mellon, protegido pelos direitos autorais, for incluído em materiais de treinamento ou num produto, o qual será vendido ou sublicenciado, a Fundação CPqD será responsável a impor todas as cláusulas necessárias para o cumprimento das obrigações referentes a esse Acordo, especificamente incluindo, mas não limitado a, à cláusula “Da Garantia” e a outras condições estabelecidas neste Acordo.

A permissão é concedida em termos de não exclusividade para propósitos comerciais.

Das Responsabilidades

Permissão adicional precisa ser obtida por escrito do Instituto de Engenharia de Software (SEI) para qualquer outro uso da Propriedade Intelectual do Instituto de Engenharia de Software da Universidade Carnegie Mellon.

A exatidão e interpretação dessa tradução são de responsabilidade da Fundação CPqD. O Instituto de Engenharia de Software e a Universidade Carnegie Mellon não têm qualquer participação nessa tradução.

DISCLAIMER

“Special permission to reproduce material on the Capability Maturity Model® (CMM®) copyrighted by Carnegie Mellon University is granted by Carnegie Mellon University.

® Capability Maturity Model is a registered mark of Carnegie Mellon University.

® CMM is registered in the U.S. Patent and Trademark Office.

“NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN “AS IS” BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

“Hold Harmless. Fundação CPqD holds CMU harmless for all liabilities, demands, damages, expenses and losses arising out of the negligent use or willful misuse of or negligent conduct or willful misconduct by Fundação CPqD regarding CMU Intellectual Property. This provision shall survive the termination of this agreement.

“Disputes. This Agreement shall be governed by the laws of the Commonwealth of Pennsylvania. Any dispute or claim arising out of or relating to this Agreement will be settled by arbitration in Pittsburgh, Pennsylvania in accordance with the rules of the American Arbitration Association and judgment upon award rendered by the arbitrator(s) may be entered in any court having jurisdiction.

“If the CMU copyrighted material is included in training matter or a product which is sold or sublicensed, Fundação CPqD is responsible to impose all provisions necessary to fulfill its obligations under this agreement, including, but not limited to the “No Warranty” clause and the other conditions stated in this agreement.

“This permission is granted on a non-exclusive basis for commercial purposes.

“Additional permission must be obtained in writing from the SEI for any other use of CMU/SEI intellectual property”.

Disclaimer

“Accuracy and interpretation of this translation are the responsibility of Fundação CPqD. The Software Engineering Institute and Carnegie Mellon University has not participated in this translation.

1. Gestão de Requisitos

Área-chave de processo para o Nível 2: Repetível

O objetivo da Gestão de Requisitos é estabelecer um entendimento comum entre o cliente e o projeto de software no qual os requisitos do cliente serão utilizados como referência.

A Gestão de Requisitos compreende o estabelecimento e a manutenção de um acordo com o cliente em relação aos requisitos do projeto de software. Este acordo é referenciado como “requisitos de sistema alocados ao software”. “Cliente” pode ser interpretado como sendo a equipe de desenvolvimento de sistema, a equipe de marketing, uma outra organização interna ou um cliente externo. O acordo cobre tanto os requisitos técnicos como os requisitos não técnicos (ex. dados do âmbito do cliente). O acordo estabelece as bases para a estimativa, o planejamento, a realização e o rastreamento das atividades do projeto de software ao longo do ciclo de vida.

A alocação dos requisitos de sistema ao software, ao hardware e a outros componentes do sistema (ex. humanos) pode ser realizada por uma equipe independente da equipe de desenvolvimento de software (ex. grupo de engenharia de sistemas), sendo que a equipe de desenvolvimento de software pode, inclusive, nem ter controle direto sobre essa alocação. Dentro das restrições de projeto, a equipe de desenvolvimento de software estabelece os passos necessários para garantir que os requisitos de sistema alocados ao software, pelos quais é responsável, sejam documentados e controlados.

Para alcançar esse controle, a equipe de desenvolvimento de software revisa os requisitos de sistema alocados ao software (os iniciais e os anteriormente já revistos), para solucionar questões antes que eles sejam incorporados ao projeto de software. Sempre que os requisitos de sistema alocados ao software são alterados, os planos de software afetados, os produtos de software e as atividades de software são ajustados para manter a consistência com os requisitos modificados.

Metas

Meta 1 Os requisitos de sistema alocados ao software são controlados para estabelecer uma linha mestra para o desenvolvimento de software e para o uso gerencial.

Meta 2 Os planos, os produtos e as atividades de software são mantidos consistentes com os requisitos de sistema alocados ao software.

Compromissos

Comprom. 1 O projeto segue uma política estabelecida pela organização para o gerenciamento de requisitos de sistema alocados ao software.

Os requisitos do sistema alocados ao software são referenciados nestas práticas como “requisitos alocados”.

Os requisitos alocados são um subconjunto dos requisitos de sistema que serão implementados em componentes de software. Os requisitos alocados constituem o principal input para o plano de desenvolvimento de software. A análise dos requisitos de software trabalha e refina os requisitos alocados, dando origem aos requisitos de software documentados.

Essa política tipicamente especifica que:

1. Os requisitos alocados são documentados.
2. Os requisitos alocados são revisados por:
 - gerentes de software,
 - outros grupos afetados.

Exemplos de grupos afetados:

- teste de sistema,
- desenvolvimento de software (incluindo todos os subgrupos, como por ex. projeto de software),
- engenharia de sistema,
- garantia da qualidade de software,
- gestão de configuração de software,
- suporte à documentação.

3. Os planos, os produtos e as atividades de software são modificados para se tornarem consistentes com as alterações dos requisitos alocados.

Habilidades

Habilidade 1

Para cada projeto, é definida a responsabilidade pelos requisitos de sistema e a alocação dos mesmos ao hardware, ao software e aos outros componentes do sistema.

A análise e a alocação dos requisitos de sistema não é responsabilidade da equipe de desenvolvimento de software, mas sim um pré-requisito para o seu trabalho.

Essa responsabilidade cobre:

1. O gerenciamento, a documentação e a alocação dos requisitos de sistema ao longo da vida do projeto.
2. A efetivação de alterações dos requisitos de sistema e a alocação dos mesmos.

Habilidade 2 Os requisitos alocados são documentados.

Os requisitos alocados incluem:

1. Os requisitos não técnicos (por ex. os acordos, as condições, e/ou os termos contratuais) que afetam e determinam as atividades de projeto de software.

Exemplos de acordos, condições e termos contratuais:

- produtos a serem entregues aos clientes,
- datas de entrega,
- marcos.

2. Os requisitos técnicos para o software.

Exemplos de requisitos técnicos:

- usuário final, operador, suporte ou funções de integração,
- requisitos de desempenho,
- restrições de projeto,
- linguagem de programação,
- requisitos de interface.

3. O critério de aceitação que será utilizado para verificar se os produtos de software satisfazem os requisitos alocados.

Habilidade 3 Recursos e orçamentos adequados são providos para a gerência dos requisitos alocados.

1. São designadas pessoas que têm experiência e perícia no domínio da aplicação e em desenvolvimento de software para gerenciar os requisitos alocados.
2. São disponibilizadas ferramentas para suportar as atividades de gestão de requisitos.

Exemplos de ferramentas de suporte:

- planilhas eletrônicas,
- ferramentas para gestão de configuração,
- ferramentas para rastreamento,
- ferramentas para gestão de testes.

Habilidade 4 Os membros da equipe de desenvolvimento de software e outros grupos de software relacionados são treinados para realizar suas atividades de gestão de requisitos.

Exemplos de treinamentos:

- métodos, padrões e procedimentos utilizados no projeto,
- domínio da aplicação.

Atividades

Atividade 1 A equipe de desenvolvimento de software revisa os requisitos alocados antes dos mesmos serem incorporados ao projeto de software.

1. São identificados os requisitos alocados que estão incompletos e que estão faltando.
2. Os requisitos alocados são revisados para determinar se os mesmos são:
 - factíveis e apropriados de serem implementados por software,
 - clara e apropriadamente enunciados,
 - consistentes entre si,
 - testáveis.
3. Quaisquer requisitos identificados como apresentando problemas em potencial são revisados juntamente com a equipe responsável pela análise e alocação dos requisitos do sistema. São realizadas as alterações necessárias.
4. Os compromissos resultantes dos requisitos alocados são negociados com os grupos afetados.

Exemplos de grupos afetados:

- desenvolvimento de software (incluindo todos os subgrupos como, por exemplo, projeto de software),
- estimativa de software,
- engenharia de sistema,
- teste de sistema,

- garantia da qualidade de software,
- gestão de configuração de software,
- gestão de contrato,
- suporte à documentação.

Vide a *Atividades 6* da área-chave de processo *Planejamento de Projeto de Software* para acessar as práticas que cobrem os compromissos da negociação.

Atividade 2

A equipe de desenvolvimento de software deve utilizar os requisitos alocados como base para os planos, para os produtos e para as atividades de software.

Os requisitos alocados:

1. São gerenciados e controlados.

“Gerenciado e controlado” implica que a versão do produto em uso em um determinado momento (passado ou presente) é conhecida (ou seja, com controle de versão), e as alterações são incorporadas de forma controlada (ou seja, com controle de alterações).

Se um grau de controle maior que o “gerenciado e controlado” é desejado, então o produto pode ser colocado sob uma forte disciplina de gestão de configuração, como descrito na área-chave de processo *Gestão de Configuração de Software*.

2. São a base para o plano de desenvolvimento de software.
3. São a base para o desenvolvimento dos requisitos de software.

Atividade 3

As modificações nos requisitos alocados são revisadas e incorporadas ao projeto de software.

1. O impacto nos compromissos existentes é avaliado e as alterações são negociadas apropriadamente.

- As modificações em compromissos com pessoas ou equipes externas à organização são revisadas com a gerência superior.

As práticas que tratam de compromissos realizados externamente à organização são encontradas na *Atividade 4* da área-chave de processo *Planejamento do Projeto de Software* e na *Atividade 3* da área-chave de processo *Acompanhamento e Supervisão de Projeto de Software*.

- As modificações em compromissos internos à organização são negociadas com os grupos afetados.

As práticas que tratam da negociação de modificações em compromissos são encontradas nas *Atividades 5, 6, 7 e 8* da área-chave de processo *Acompanhamento e Supervisão de Projeto de Software*.

2. As alterações necessárias nos planos, nos produtos e nas atividades de software resultantes das modificações dos requisitos alocados, são:
 - identificadas,
 - avaliadas,
 - avaliadas quanto a riscos,
 - documentadas,
 - planejadas,
 - comunicadas às equipes e pessoas afetadas,
 - acompanhadas até a sua conclusão.

Medições e Análises

Medição 1

As medições são realizadas e utilizadas para determinar a situação das atividades de gestão de requisitos alocados.

Exemplos de medições:

- situação de cada um dos requisitos alocados,
- atividade de alteração dos requisitos alocados,
- número acumulado de alterações nos requisitos alocados, incluindo o número de alterações propostas, abertas, aprovadas e incorporadas às linhas básicas do sistema.

Verificação da Implementação

Verificação 1

As atividades de gestão de requisitos alocados são revisadas periodicamente por um gerente superior.

A finalidade básica das revisões periódicas pelo gerente superior é fornecer regularmente ciência e compreensão sobre as atividades do processo de software a um nível de abstração adequado. O intervalo de tempo entre as revisões deveria ajustar-se às necessidades da organização, podendo ser prolongado à medida que mecanismos adequados para encaminhamento de exceções estejam disponíveis.

Vide a *Verificação 1* da área-chave de processo *Acompanhamento e Supervisão de Projeto de Software* para acessar as práticas que cobrem o conteúdo típico da supervisão de revisões pela gerência superior.

Verificação 2

As atividades de gestão de requisitos alocados são revisadas pelo gerente de projeto, tanto periodicamente, como motivadas por um evento.

Vide a *Verificação 2* da área-chave de processo *Acompanhamento e Supervisão de Projeto de Software* para acessar as práticas que cobrem o conteúdo típico da supervisão de revisões pela gestão do projeto.

Verificação 3 **A equipe de garantia da qualidade de software deve revisar e/ou proceder à auditoria das atividades e dos produtos da gestão de requisitos alocados e reportar os resultados.**

Vide a área-chave de processo Garantia da Qualidade de Software.

No mínimo, essas revisões e auditorias devem verificar se:

1. Os requisitos alocados foram revisados e os problemas foram resolvidos antes da equipe de desenvolvimento de software comprometer-se com eles.
2. Os planos, os produtos e as atividades de software foram apropriadamente revisados quando os requisitos alocados sofreram modificações.
3. As alterações em compromissos, resultantes de modificações nos requisitos alocados, foram negociadas com os grupos afetados.

2. Planejamento de Projeto de Software

Área-chave de processo para o Nível 2: Repetível

O objetivo do Planejamento de Projeto de Software é estabelecer planos razoáveis para a execução das atividades de engenharia de software e para a gestão do projeto de software.

O Planejamento de Projeto de Software compreende a elaboração de estimativas para o trabalho a ser realizado, o estabelecimento de compromissos necessários e a definição de planos para a realização do trabalho.

O Planejamento de Projeto de Software se inicia com o estabelecimento do trabalho a ser realizado e de outras restrições e metas que definem e limitam o escopo do projeto de software (aquelas estabelecidas pelas práticas da área-chave de processo *Gestão de Requisitos*). O processo de planejamento de software inclui passos para estimar o tamanho dos produtos de software e os recursos necessários, para produzir um cronograma, para identificar e avaliar riscos e para negociar compromissos. Devido a iterações destes passos, pode ser necessário estabelecer um plano para o projeto de software (ou seja, o plano de desenvolvimento de software).

Esse plano fornece as bases para execução e gestão das atividades do projeto de software e explicita os compromissos do cliente, de acordo com os recursos, as restrições e as capacidades do projeto de software.

Metas

- Meta 1** As estimativas de software são documentadas para serem utilizadas no planejamento e no acompanhamento de projeto.
- Meta 2** As atividades e os compromissos do projeto de software são planejados e documentados.
- Meta 3** Os grupos e as pessoas afetadas estão de acordo com os seus compromissos relacionados ao projeto de software.

Compromissos

- Comprom. 1** Um gerente de projeto de software é designado para ser o responsável pela negociação dos compromissos e pela elaboração do plano de desenvolvimento de software do projeto.
- Comprom. 2** O projeto segue uma política organizacional formal (escrita) para o planejamento de projeto de software.

Essa política tipicamente especifica que:

1. Os requisitos de sistema alocados ao software são utilizados como base para o planejamento de projeto de software.

Vide a *Atividade 2* da área-chave de processo *Gestão de Requisitos*.

2. Os compromissos do projeto de software são negociados entre:
 - gerente do projeto,
 - gerente de software do projeto,
 - os outros gerentes de software.

3. O envolvimento de outras equipes de desenvolvimento nas atividades de software é negociado com essas equipes e é documentado.

Exemplos de outras equipes:

- engenharia de sistema,
- hardware,
- teste de sistema.

4. Os grupos afetados revisam:

- as estimativas de tamanho do software,
- as estimativas de esforço e de custo do projeto de software,
- os cronogramas,
- outros compromissos.

Exemplos de grupos afetados:

- desenvolvimento de software (incluindo todos os subgrupos, por ex, grupo de arquitetura),
- estimativa de software,
- engenharia de sistema,
- teste de sistema,
- garantia da qualidade de software,
- gestão de contrato,
- suporte à documentação.

5. O gerente superior revisa todos os compromissos do projeto de software assumidos com pessoas e grupos externos à organização.

6. O plano de desenvolvimento de software do projeto é gerenciado e controlado.

O termo “plano de desenvolvimento de software” é utilizado nestas práticas para referir-se ao plano global para gestão do projeto de software. O uso da terminologia “desenvolvimento” não tem a intenção de excluir a manutenção de software ou projetos de suporte e deveria ser interpretado apropriadamente no contexto do projeto.

“Gerenciado e controlado” implica que a versão do produto em uso em um determinado momento (passado ou presente) é conhecida (ou seja, com controle de versão), e as alterações são incorporadas de forma controlada (ou seja, com controle de alterações). Se um grau de controle maior que o “gerenciado e controlado” é desejado, então o produto pode ser colocado sob uma forte disciplina de gestão de configuração, como descrito na área-chave de processo *Gestão de Configuração de Software*.

Habilidades

Habilidade 1

Existe um contrato de projeto de software documentado e aprovado para o projeto de software.

1. O contrato de projeto de software deve cobrir:

- escopo do trabalho,
- os objetivos e as metas técnicas,
- a identificação dos clientes e dos usuários,

Os usuários finais referenciados nestas práticas são os usuários finais designados pelos clientes ou representantes dos usuários finais.

- os padrões impostos,
- as responsabilidades atribuídas,
- as metas e as restrições de custo e de cronograma,
- as dependências entre o projeto de software e outras organizações,

Exemplos de outras organizações:

- cliente,
- subcontratados,
- parceiros de *joint venture*.

- as metas e as restrições de recursos,
- outras restrições e objetivos para desenvolvimento e/ou manutenção.

2. O contrato de projeto de software é revisado pelo:
 - gerente do projeto,
 - gerente do projeto de software,
 - outros gerentes de software,
 - outros grupos afetados.
3. O contrato de projeto de software é gerenciado e controlado.

Habilidade 2**As responsabilidades para a elaboração do plano de desenvolvimento de software são designadas.**

1. O gerente do projeto de software coordena, diretamente ou através de delegação, o planejamento do projeto de software.
2. As responsabilidades pelos produtos de software e pelas atividades de software são particionadas e atribuídas aos gerentes de software, de forma rastreável e contabilizável.

Exemplos de produtos de software:

- produtos a serem entregues ao cliente externo ou ao usuário final,
- produtos a serem utilizados por outras equipes de desenvolvimento,
- produtos a serem utilizados pela equipe de desenvolvimento.

Habilidade 3**Recursos e orçamento adequados são providos para o planejamento de projeto de software.**

1. Sempre que factível, pessoas experientes, que possuem conhecimento no domínio da aplicação do projeto de software que está sendo planejado, devem estar disponíveis para elaborar o plano de desenvolvimento de software.
2. As ferramentas para suportar as atividades de planejamento de projeto são disponibilizadas.

Exemplos de ferramentas de suporte:

- planilhas eletrônicas,
- modelos de estimativas,
- programas de planejamento de projeto e de elaboração de cronogramas.

Habilidade 4 Os gerentes de software, engenheiros de software e outras pessoas envolvidas no planejamento do projeto de software são treinados em estimativas de software e procedimentos de planejamento aplicáveis às suas áreas de responsabilidade.

Atividades

Atividade 1 A equipe de desenvolvimento de software participa da elaboração da proposta do projeto.

1. A equipe de desenvolvimento de software é envolvida na:
 - preparação da proposta e sua submissão,
 - clarificação de discussões e suas submissões,
 - negociações de alterações em compromissos que afetam o projeto de software.
2. A equipe de desenvolvimento de software deve revisar os compromissos propostos do projeto.

Exemplos de compromissos de projeto:

- metas técnicas e objetivos do projeto,
- soluções técnicas de software e de sistema,
- orçamento, cronograma e recursos para o software,
- padrões e procedimentos para o software.

Atividade 2 O planejamento de projeto de software é iniciado nos primeiros estágios e em paralelo com o planejamento geral do projeto.

Atividade 3 A equipe de desenvolvimento de software, junto com outros grupos afetados, participa do planejamento geral durante todo o projeto.

1. A equipe de desenvolvimento de software revisa os planos das várias fases do projeto.

Atividade 4 Os compromissos do projeto de software assumidos com pessoas e grupos externos à organização são revisados em conjunto com a gerência superior, de acordo com um procedimento formal (documentado).

Atividade 5 O ciclo de vida de software, com fases pré-definidas de tamanho gerenciável, é identificado ou definido.

Exemplos de ciclos de vida de software:

- cascata,
- cascata sobreposta,
- espiral,
- serial,
- protótipo único/cascata sobreposta.

Atividade 6 O plano de desenvolvimento de software é elaborado de acordo com um procedimento formal (documentado).

Esse procedimento tipicamente especifica que:

1. O plano de desenvolvimento de software é baseado em:
 - padrões do cliente (quando apropriado),
 - padrões do projeto,
 - contratos de projeto de software aprovados,
 - requisitos alocados ao software.
2. Os planos para as equipes relacionadas ao software e outras equipes de desenvolvimento envolvidas nas atividades da equipe de desenvolvimento de software são negociados com essas equipes, os esforços de suporte são orçados e os acordos são documentados.

Exemplos de equipes relacionadas ao software:

- garantia da qualidade de software,
- gestão de configuração de software,
- suporte à documentação

Exemplos de equipes de desenvolvimento:

- desenvolvimento de sistema,
- desenvolvimento de hardware,
- teste de sistema.

3. Os planos para envolvimento da equipe de desenvolvimento de software nas atividades de outras equipes relacionadas ao software e outras equipes de desenvolvimento são negociados com aquelas equipes, os esforços de suporte são orçados e os acordos são documentados.
4. O plano de desenvolvimento de software é revisado pelo:
 - gerente do projeto,
 - gerente do projeto de software,
 - outros gerentes de software,
 - outros grupos afetados.
5. O plano de desenvolvimento de software é gerenciado e controlado.

Atividade 7

O plano para o projeto de software é documentado.

Nas práticas-chave, refere-se a esse plano, ou conjunto de planos, como plano de desenvolvimento de software.

Vide a *Verificação 1* da área-chave de processo *Acompanhamento e Supervisão de Projeto de Software* para acessar as práticas relativas à utilização do plano de desenvolvimento de software.

O plano de desenvolvimento abrange:

1. O propósito, o escopo, as metas e os objetivos do projeto de software.
2. A seleção do ciclo de vida do software.
3. A identificação dos procedimentos, dos métodos e dos padrões para desenvolvimento e/ou manutenção de software.

Exemplos de procedimentos e padrões:

- planejamento de desenvolvimento de software,
- gestão de configuração de software,
- garantia da qualidade de software,
- projeto de software,
- acompanhamento e resolução de problemas,
- medição de software.

4. A identificação dos produtos de software a serem desenvolvidos.
5. As estimativas do tamanho dos produtos de software a serem desenvolvidos.
6. As estimativas de esforços e de custo do projeto de software.
7. O uso estimado de recursos computacionais críticos.
8. Os cronogramas do projeto de software, incluindo a identificação de marcos e revisões.
9. A identificação e avaliação dos riscos do projeto de software.
10. O plano de facilidades de engenharia de software e de ferramentas de suporte para o projeto.

Atividade 8

Os produtos de software necessários para estabelecer e manter o controle do projeto de software são identificados.

Vide a Atividade 4 da área-chave de processo Gestão de Configuração de Software.

Atividade 9

As estimativas de tamanho dos produtos de software (ou mudanças no tamanho dos produtos de software) são obtidas de acordo com um procedimento formal (documentado).

Esse procedimento tipicamente especifica que:

1. As estimativas de tamanho são feitas para todas as principais atividades de software e produtos de software.

Exemplos de medições de tamanho de software:

- pontos por função,
- features points,
- linhas de código,
- número de requisitos,
- número de páginas.

Exemplos de tipos de produtos e atividades para os quais as estimativas de tamanho são feitas:

- sistema operacional e software de suporte,
- produtos que devem e que não devem ser entregues aos clientes/usuários,
- produtos de software e produtos que não são de software (ex: documentos),
- atividades de desenvolvimento, verificação e validação de produtos.

2. Os produtos de software são decompostos até a granularidade necessária para se alcançar os objetivos das estimativas.
3. Os dados históricos são utilizados quando disponíveis.
4. As suposições utilizadas na estimativa de tamanho são documentadas.
5. As estimativas de tamanho são documentadas, revisadas e consensadas.

Exemplos de grupos e pessoas que revisam e consensam as estimativas de tamanho:

- gerente de projeto,
- gerente de projeto de software,
- outros gerentes de software.

Atividade 10

As estimativas de esforço e de custos do projeto de software são obtidas de acordo com um procedimento formal (documentado).

Esse procedimento tipicamente especifica que:

1. As estimativas de esforço e de custo do projeto de software estão relacionadas com as estimativas de tamanho dos produtos de software (ou do tamanho das mudanças).
2. Os dados de produtividade (históricos ou atuais) são utilizados nas estimativas quando disponíveis, as fontes e as bases lógicas para esses dados são documentadas.
 - os dados de produtividade e de custo são obtidos, sempre que possível, na organização em que o projeto está sendo desenvolvido.
 - os dados de produtividade e de custo devem levar em conta o esforço e os custos significativos da construção dos produtos de software.

Exemplos de custos significativos de construção dos produtos de software:

- despesas com mão de obra direta,
- despesas adicionais,
- despesas com viagens,
- despesas com utilização de computadores.

3. As estimativas de custo, de esforço, e de pessoal são baseadas em experiências anteriores.
 - projetos similares deveriam ser utilizados quando possível,
 - a determinação das atividades e suas distribuições no tempo são derivadas,
 - as estimativas de esforço, de pessoal e de custo ao longo do ciclo de vida do software são preparadas.
4. As estimativas e as suposições feitas na obtenção das estimativas são documentadas, revisadas e consensadas.

Atividade 11

As estimativas de recursos computacionais críticos para o projeto são obtidas de acordo com um procedimento formal (documentado).

Os recursos computacionais críticos podem estar no ambiente do *host*, no ambiente de teste e integração, no ambiente alvo, ou em qualquer combinação destes.

Esse procedimento tipicamente especifica que:

1. Os recursos computacionais críticos para o projeto são identificados.

Exemplos de recursos computacionais críticos:

- capacidade de memória do computador,
- utilização do processador do computador,
- capacidade do canal de comunicação.

2. As estimativas dos recursos computacionais críticos devem estar relacionadas com as estimativas de:
 - tamanho dos produtos de software,
 - carga do processamento operacional,
 - tráfego de comunicação.
3. As estimativas dos recursos computacionais críticos são documentadas, revisadas e consensadas.

Atividade 12

O cronograma do projeto de software é obtido de acordo com um procedimento formal (documentado).

Esse procedimento tipicamente especifica que:

1. O cronograma de software está relacionado com:
 - a estimativa de tamanho dos produtos de software (ou do tamanho das mudanças),
 - esforço e o custo do software.
2. O cronograma de software é baseado em experiências anteriores.
 - projetos similares são utilizados quando possível.

3. O cronograma do software deve acomodar os marcos (datas) impostos, datas com dependências críticas e outras restrições.
4. As atividades do cronograma de software têm duração apropriada e os marcos são espaçados apropriadamente no tempo para permitir precisão nas medições de progresso das atividades.
5. As suposições feitas na elaboração do cronograma são documentadas.
6. O cronograma de software é documentado, revisado e consensado.

Atividade 13

Os riscos de software associados a custo, recursos, cronograma e aspectos técnicos do projeto são identificados, avaliados e documentados.

1. Os riscos são analisados e atribuídas prioridades com base em seus impactos potenciais no projeto.
2. As contingências para os riscos são identificadas.

Exemplos de contingências:

- folgas no cronograma,
- planos alternativos para o quadro de funcionários,
- planos alternativos para equipamentos computacionais adicionais.

Atividade 14

Os planos de facilidades de desenvolvimento de software do projeto e as ferramentas de suporte são preparados.

1. As estimativas de requisitos de capacidade para essas facilidades e ferramentas de suporte são baseadas nas estimativas de tamanho dos produtos de software e outras características.

Exemplos de facilidades de desenvolvimento de software e de ferramentas de suporte:

- computadores e periféricos para desenvolvimento de software, computadores e periféricos para teste de software,
- software de ambiente do computador alvo,
- outros softwares de suporte.

2. As responsabilidades são atribuídas e os compromissos são negociados para obter ou desenvolver essas facilidades e ferramentas de suporte.
3. Os planos são revisados por todos os grupos afetados.

Atividade 15

Os dados de planejamento de software são registrados.

1. A informação registrada inclui as estimativas e as informações associadas necessárias à reconstrução das estimativas e à avaliação de quão são razoáveis.
2. Os dados de planejamento de software são gerenciados e controlados.

Medição e Análises

Medição 1

As medições são feitas e utilizadas para determinar a situação das atividades do planejamento de software.

Exemplos de medições:

- alcance dos marcos para as atividades de planejamento de projeto de software em relação ao plano,
- trabalhos concluídos, esforço e orçamento despendidos nas atividades de planejamento de projeto de software em relação ao plano.

Verificação da Implementação

Verificação 1 **As atividades de planejamento de projeto de software são revisadas periodicamente por um gerente superior.**

A finalidade básica das revisões periódicas pelo gerente superior é fornecer regularmente informações sobre as atividades do processo de software em um nível de abstração adequado. O intervalo de tempo entre as revisões deveria ajustar-se às necessidades da organização, podendo ser prolongado na medida em que estejam disponíveis mecanismos adequados para encaminhamento de exceções.

1. O desempenho técnico, de orçamento, de pessoal e de cronograma são revisados.
2. Os conflitos que não podem ser resolvidos nos níveis inferiores são encaminhados.
3. Os riscos do projeto de software são encaminhados.
4. Os itens de ação são atribuídos, revisados e acompanhados até sua conclusão.
5. Um relatório resumido de cada reunião é preparado e distribuído aos grupos e pessoas afetadas.

Verificação 2 **As atividades de planejamento de projeto de software são revisadas pelo gerente de projeto, periodicamente ou motivado por um evento.**

1. Os grupos afetados devem estar representados.
2. A situação e os resultados correntes das atividades do planejamento de projeto de software são revisados com relação ao contrato de projeto de software e aos requisitos alocados.
3. As dependências entre grupos são encaminhadas.
4. Os conflitos não solucionáveis nos níveis inferiores são encaminhados.
5. Os riscos do projeto de software são revisados.

6. Os itens de ação são atribuídos, revisados e acompanhados até a sua conclusão.
7. Um relatório resumido de cada reunião é preparado e distribuído às pessoas e grupos afetados.

Verificação 3

A equipe de Garantia da Qualidade de Software deve revisar e/ou realizar a auditoria das atividades e produtos do planejamento de projeto de software e reportar seus resultados.

Vide a área-chave de processo *Garantia de Qualidade de Software*.

No mínimo, essas revisões e auditorias devem verificar:

- as atividades de planejamento e estimativas de software,
- as atividades de revisão e execução dos compromissos do projeto,
- as atividades de preparação do plano de desenvolvimento de software,
- os padrões utilizados na preparação do plano de desenvolvimento de software,
- conteúdo do plano de desenvolvimento de software.

3. Acompanhamento e Supervisão de Projeto de Software

Área-chave de processo para o Nível 2: Repetível

O objetivo do Acompanhamento e Supervisão do Projeto de Software é prover visibilidade adequada do progresso real, permitindo que a gerência execute ações efetivas quando o desempenho do projeto desvia significativamente dos planos de software.

O Acompanhamento e Supervisão do Projeto de Software compreende o acompanhamento e a revisão das realizações do projeto de software e dos resultados obtidos em relação às estimativas, aos compromissos, e aos planos documentados, ajustando esses últimos às reais realizações e resultados.

Um plano documentado para o projeto de software (ou seja, o plano de desenvolvimento de software, como descrito na área-chave de processo Planejamento de Projeto de Software) é utilizado como base para acompanhamento das atividades de software, para comunicação da situação do projeto e revisão dos planos. As atividades de software são monitoradas pela gerência. O progresso é inicialmente determinado pela comparação do tamanho do software, esforço, custo e cronograma reais com o plano quando os produtos de software selecionados são concluídos e os marcos escolhidos são alcançados. Quando se observa que os planos do projeto de software não estão sendo cumpridos, ações corretivas são executadas. Essas ações podem incluir a revisão do plano de desenvolvimento de software para refletir os resultados reais alcançados e o replanejamento do trabalho restante ou tomada de medidas para melhorar o desempenho.

Metas

- Meta 1** Os resultados e o desempenho reais são rastreados com relação aos planos de software.
- Meta 2** As ações corretivas são executadas e gerenciadas até sua conclusão, quando os resultados e o desempenho reais desviam significativamente dos planos de software.
- Meta 3** As alterações nos compromissos de software são consensadas pelos indivíduos e pelos grupos afetados.

Compromissos

- Comprom. 1** Um gerente de projeto de software deve ser designado para ser o responsável pelas atividades e resultados do projeto de software.
- Comprom. 2** O projeto deve seguir uma política organizacional formal (escrita) para o gerenciamento do projeto de software.

Esta política tipicamente especifica que:

1. Um plano de desenvolvimento de software documentado é utilizado e mantido como base para acompanhamento do projeto de software.
2. O gerente do projeto é mantido informado sobre a situação e questões do projeto de software.
3. As ações corretivas são tomadas quando o plano de software não está sendo alcançado, através de ajustes no desempenho ou ajustes nos planos.
4. As alterações nos compromissos de software são realizadas com o envolvimento e a concordância dos grupos afetados.

Exemplos de grupos afetados:

- desenvolvimento de software (incluindo todos os subgrupos, como projeto de software),
- estimativa de software,
- engenharia de sistema,
- teste de sistema,
- garantia da qualidade de software,
- gestão de configuração de software,
- gestão de contrato,
- suporte à documentação.

5. O gerente superior revisa todas as alterações de compromissos e todos os novos compromissos do projeto de software assumidos com indivíduos e grupos externos à organização.

Habilidades

Habilidade 1

O plano de desenvolvimento de software para o projeto de software é documentado e aprovado.

Vide as *Atividades 6 e 7* da área-chave de processo *Planejamento de Projeto de Software* para acessar as práticas que cobrem o plano de desenvolvimento de software.

Habilidade 2

O gerente do projeto de software atribui explicitamente as responsabilidades pelos produtos e atividades de software.

As responsabilidades atribuídas cobrem:

1. Os produtos de software a serem desenvolvidos ou os serviços a serem fornecidos.
2. O esforço e o custo para essas atividades de software.
3. O cronograma para essas atividades de software.

4. O orçamento para essas atividades de software.

Habilidade 3 Os recursos e os orçamentos adequados são fornecidos para o acompanhamento do projeto de software.

1. Aos gerentes de software e aos líderes de tarefas de software são atribuídas responsabilidades específicas para o acompanhamento do projeto de software.
2. São disponibilizadas ferramentas para suportar o acompanhamento do projeto.

Exemplos de ferramentas de suporte:

- planilhas eletrônicas;
- programas de planejamento de projeto e de elaboração de cronogramas.

Habilidade 4 Os gerentes de software são treinados para gerenciar aspectos técnicos e pessoais do projeto de software.

Exemplos de treinamento:

- gestão técnica de projetos;
- acompanhamento e supervisão de tamanho, esforço, custo, e cronograma;
- gestão de pessoal.

Habilidade 5 Os gerentes de 1ª linha recebem orientação sobre os aspectos técnicos do projeto de software.

Exemplos de orientação:

- padrões e procedimentos de engenharia de software para o projeto,
- domínio da aplicação do projeto.

Atividades

Atividade 1 Um plano de desenvolvimento documentado é utilizado para acompanhar as atividades de software e comunicar sua situação.

Vide a *Atividade 7* da área-chave de processo *Planejamento de Projeto de Software* para acessar as práticas relativas ao conteúdo do plano de desenvolvimento de software.

Esse plano de desenvolvimento é:

1. Atualizado à medida que o trabalho avança para refletir o cumprimento de tarefas, particularmente quando os marcos são alcançados.
2. Disponibilizado para leitura:
 - ao grupo de desenvolvimento de software (incluindo todos os subgrupos, tal como projeto de software);
 - ao gerente do projeto,
 - ao gerente de software,
 - à gerência superior,
 - a outros grupos afetados.

Atividade 2 O plano de desenvolvimento de software do projeto é revisado de acordo com um procedimento formal (escrito).

Vide a *Atividade 6* da área-chave de processo *Planejamento de Projeto de Software* para acessar as práticas relativas às atividades de elaboração do plano de desenvolvimento de software.

Esse procedimento tipicamente especifica que:

1. O plano de desenvolvimento de software é revisado, quando apropriado, para incorporar alterações e refinamentos no mesmo, particularmente quando os planos mudam significativamente.

As interdependências entre os requisitos alocados ao software, as restrições de projeto, os recursos, os custos e os cronogramas precisam ser refletidas em todas as alterações realizadas no plano.

2. O plano de desenvolvimento de software é atualizado para incorporar todos os novos compromissos do projeto de software e as alterações nos compromissos já existentes.
3. O plano de desenvolvimento de software é revisto a cada revisão.
4. O plano de desenvolvimento de software é gerenciado e controlado.

“Gerenciado e controlado” implica que a versão do produto em uso em um determinado momento (passado ou presente) é conhecida (ou seja, com controle de versão), e as alterações são incorporadas de forma controlada (ou seja, com controle de alterações).

Se um grau de controle maior que o “gerenciado e controlado” é desejado, então o produto pode ser colocado sob a disciplina de gestão de configuração, como descrito na área-chave de processo *Gestão de Configuração de Software*.

Atividade 3

Os compromissos do projeto de software e as alterações de compromissos assumidos com indivíduos e equipes externas à organização são revisados em conjunto com a gerência superior, de acordo com um procedimento formal (escrito).

Atividade 4

As alterações de compromissos aprovadas que afetam o projeto de software são comunicadas aos membros da equipe de desenvolvimento e às outras equipes de software relacionadas.

Exemplos de outras equipes de software relacionadas:

- garantia da qualidade de software,
- gestão de configuração de software,
- suporte à documentação.

Atividade 5 O tamanho dos produtos de trabalho de software (ou o tamanho das alterações nos produtos de trabalho de software) são acompanhados e ações corretivas são executadas quando necessário.

Vide a *Atividade 9* da área-chave de processo *Planejamento de Projeto de Software* para acessar as práticas que cobrem a elaboração de estimativas de tamanho.

1. O tamanho de todos os produtos de trabalho de software mais importantes (ou tamanho das alterações) são acompanhados.
2. O tamanho real do código (gerado, totalmente testado e entregue) é comparado com as estimativas documentadas no plano de desenvolvimento de software.
3. As unidades reais de documentação entregues ao cliente são comparadas com as estimativas documentadas no plano de desenvolvimento de software.
4. O tamanho projetado dos produtos de trabalho de software (estimativas combinadas com os valores reais) é refinado, monitorado e ajustado como um todo em bases regulares.
5. As alterações nas estimativas do tamanho dos produtos de trabalho de software que afetam os compromissos de software são negociadas com os grupos afetados e são documentadas.

Atividade 6 O esforço e os custos do projeto de software são acompanhados e ações corretivas são executadas quando necessário.

Vide a *Atividade 10* da área-chave de processo *Planejamento de Projeto de Software* para acessar as práticas que cobrem a elaboração de estimativas de custo.

1. Os dispêndios reais de esforço e de custos de horas extras e de trabalhos completados são comparados com as estimativas documentadas no plano de desenvolvimento de software para identificar potenciais estimativas a maior e a menor.

2. Os custos de software são acompanhados e comparados com as estimativas documentadas no plano de desenvolvimento de software.
3. O esforço e o quadro de pessoal são comparados com as estimativas documentadas no plano de desenvolvimento de software.
4. As alterações no quadro de pessoal e outros custos de software que afetam os compromissos de software são negociados com os grupos afetados e são documentados.

Atividade 7

Os recursos computacionais críticos do projeto são acompanhados e ações corretivas são executadas quando necessário.

Vide a *Atividade 11* da área-chave de processo *Planejamento de Projeto de Software* para acessar as práticas que cobrem a elaboração de estimativas de recursos computacionais.

1. A utilização real e a utilização projetada dos recursos computacionais críticos do projeto são acompanhadas e comparadas com as estimativas feitas para cada um dos principais componentes de software, como documentado no plano de desenvolvimento de software.
2. As alterações nas estimativas dos recursos computacionais críticos que afetam os compromissos de software são negociadas com os grupos afetados e são documentadas.

Atividade 8

O cronograma de software do projeto é acompanhado e ações corretivas são executadas quando necessário.

Vide a *Atividade 12* da área-chave de processo *Planejamento de Projeto de Software* para acessar as práticas que cobrem a elaboração de cronograma.

1. A conclusão efetiva das atividades de software, o alcance de marcos e outros compromissos são confrontados com o plano de desenvolvimento de software.

2. Os efeitos de atrasos e adiantamentos na conclusão das atividades de software, marcos e outros compromissos são avaliados com relação a impactos em futuras atividades e marcos.
3. As revisões do cronograma de software que afetam os compromissos de software são negociadas com os grupos afetados e são documentadas.

Atividade 9 **As atividades técnicas de desenvolvimento de software são acompanhadas e ações corretivas são realizadas quando necessário.**

1. Os membros da equipe de desenvolvimento de software reportam regularmente sua situação técnica para o gerente imediato.
2. Os conteúdos das “releases” do software, para as sucessivas versões, são comparados com os planos documentados no plano de desenvolvimento de software.
3. Os problemas identificados em quaisquer produtos de trabalho de software são reportados e documentados.
4. Os relatórios de problemas são acompanhados até a sua conclusão.

Atividade 10 **Os riscos de software associados a custo, recurso, cronograma e aspectos técnicos do projeto são acompanhados.**

Vide a *Atividade 13* da área-chave de processo *Planejamento de Projeto de Software* para acessar as práticas que cobrem a identificação de riscos.

1. As prioridades dos riscos e as contingências para os riscos são ajustadas quando informações adicionais tornam-se disponíveis.
2. As áreas de alto risco são revisadas regularmente com o gerente do projeto.

Atividade 11 Os dados reais de medição e os dados de replanejamento do projeto de software são registrados.

Vide a *Atividade 15* da área-chave de processo *Planejamento de Projeto de Software* para acessar as práticas que cobrem o registro de dados do projeto.

1. As informações registradas incluem as estimativas e as informações associadas necessárias para reconstruir as estimativas e verificar o quão são razoáveis.
2. Os dados de replanejamento de software são gerenciados e controlados.
3. Os dados de planejamento de software, os dados de replanejamento e os dados reais de medições são arquivados para uso em projetos futuros e em andamento.

Atividade 12 O grupo de desenvolvimento de software conduz revisões internas periódicas para acompanhar o progresso técnico, os planos, o desempenho e as questões relacionadas ao plano de desenvolvimento de software.

Essas revisões são conduzidas entre:

1. O gerente de software (de 1^a linha) e seus líderes de tarefas de software.
2. O gerente do projeto de software, os gerentes de software (de 1^a linha) e outros gerentes de software, quando apropriado.

Atividade 13 As revisões formais para encaminhar as realizações e resultados do projeto de software são conduzidas em marcos selecionados do projeto de acordo com um procedimento formal (escrito).

Essas revisões:

1. São planejadas para ocorrer em pontos significativos do cronograma do projeto de software, tais como o início ou a conclusão de determinadas etapas.
2. São conduzidas com o cliente, com o usuário final e com os grupos afetados dentro da organização.

Os usuários finais referidos nestas práticas são os usuários finais designados pelo cliente ou representantes dos usuários finais.

3. Utilizam materiais que são revisados e aprovados pelos gerentes de software responsáveis.
4. Encaminham os compromissos, os planos e a situação das atividades de software.
5. Resultam na identificação e documentação de questões, itens de ação e decisões significativas.
6. Encaminham os riscos do projeto de software.
7. Resultam no refinamento do plano de desenvolvimento de software, quando necessário.

Medição e Análises

Medição 1

As medições são feitas e utilizadas para determinar a situação das atividades de acompanhamento e supervisão de software.

Exemplos de medições:

- esforço e outros recursos utilizados na execução das atividades de acompanhamento e supervisão,
- atividade de alteração no plano de desenvolvimento de software, que inclui mudanças nas estimativas do tamanho dos produtos de trabalho de software, nas estimativas de custo de software, nas estimativas de recursos computacionais críticos e nos cronogramas.

Verificação da Implementação

Verificação 1 **As atividades de acompanhamento e supervisão de projeto de software são revisadas periodicamente com um gerente superior.**

A finalidade básica das revisões periódicas pelo gerente superior é fornecer regularmente ciência e compreensão sobre as atividades do processo de software a um nível de abstração adequado. O intervalo de tempo entre as revisões deveria ajustar-se às necessidades da organização, podendo ser prolongado à medida em que mecanismos adequados para relato das exceções estejam disponíveis.

1. O desempenho técnico, de gasto, de quadro de pessoal e do cronograma são revisados.
2. Os conflitos e as questões que não podem ser resolvidos nos níveis mais baixos são encaminhados.
3. Os riscos do projeto de software são encaminhados.
4. Os itens de ação são atribuídos, revisados e acompanhados até a sua conclusão.
5. Um relatório resumido de cada reunião é preparado e distribuído aos grupos afetados.

Verificação 2 **As atividades de acompanhamento e supervisão de projeto de software são revisadas com o gerente de projeto, tanto periodicamente como motivado por evento.**

1. Os grupos afetados estão representados.
2. O desempenho técnico, de gasto, de quadro de pessoal e do cronograma são revisados com relação ao plano de desenvolvimento de software.
3. A utilização de recursos computacionais críticos é revisada; as estimativas atuais e a utilização real desses recursos computacionais críticos são reportadas com relação às estimativas originais.
4. As dependências entre os grupos são encaminhadas.

5. Os conflitos e as questões não solucionáveis nos níveis inferiores são encaminhados.
6. Os riscos do projeto de software encaminhados.
7. Os itens de ação são atribuídos, revisados e acompanhados até a sua conclusão.
8. Um relatório resumido de cada reunião é preparado e distribuído aos grupos afetados.

Verificação 3

O grupo de Garantia da Qualidade de Software revisa e/ou realiza a auditoria das atividades e dos produtos de trabalho para o acompanhamento e supervisão de projeto de software e relata seus resultados.

Vide a área-chave de processo *Garantia da Qualidade de Software*.

No mínimo, essas revisões e auditorias verificam:

1. As atividades de revisão e a execução dos compromissos.
2. As atividades de revisão do plano de desenvolvimento de software.
3. O conteúdo do plano do plano de desenvolvimento de software revisado.
4. As atividades de acompanhamento de custo, de cronograma, de riscos, de restrições técnicas e de *design* do projeto de software, bem como a funcionalidade e o desempenho.
5. As atividades de condução das revisões técnicas e gerenciais planejadas.

4. Gestão de Subcontratação de Software

Área-chave de processo para o Nível 2: Repetível

O objetivo da Gestão de Subcontratação de Software é selecionar os subcontratados de software e gerenciá-los efetivamente.

A Gestão de Subcontratação de Software compreende a seleção do subcontratado de software, estabelecendo compromissos com o subcontratado, acompanhando e revisando o desempenho e os resultados do subcontratado. Estas práticas cobrem a gestão de subcontratação de software (somente), como também a gestão dos componentes de software do subcontratado que inclui software, hardware e possivelmente outros componentes de sistema.

O subcontratado é selecionado baseado em suas habilidades de realizar o trabalho. Muitos fatores contribuem para a decisão de subcontratar uma parte do trabalho do contratante principal. Os subcontratados podem ser selecionados tanto com base em alianças estratégicas de negócio, como também com base em considerações técnicas. As práticas desta área-chave de processo tratam do processo tradicional de aquisição associado à subcontratação de uma organização para realizar uma parte definida do trabalho para outra organização.

Quando é realizada uma subcontratação, é estabelecido um acordo documentado, cobrindo requisitos técnicos e não técnicos (datas de entrega, por ex.), que é utilizado como base para gerenciar o subcontratado. O trabalho a ser feito pelo subcontratado e os planos para a realização do trabalho são documentados. Os padrões que devem ser seguidos pelo subcontratado são compatíveis com os padrões iniciais do contratante principal.

As atividades de planejamento de software e de acompanhamento e supervisão do trabalho subcontratado são realizadas pelo contratante. O contratante principal garante que essas atividades de planejamento, acompanhamento e supervisão são realizadas apropriadamente e que os produtos de software entregues pelo subcontratado satisfazem seus critérios de aceitação. O contratante principal trabalha com o subcontratado para gerenciar suas interfaces de produtos e processos.

Metas

- Meta 1** O contratante principal seleciona subcontratados de software qualificados.
- Meta 2** O contratante e o subcontratado de software concordam com os compromissos assumidos entre eles.
- Meta 3** O contratante e o subcontratado de software mantêm comunicações constantes.
- Meta 4** O contratante acompanha os resultados reais e o desempenho do subcontratado com relação aos compromissos.

Compromissos

- Comprom. 1** O projeto segue uma política organizacional formal (escrita) para a gestão do subcontrato de software.

Esta política tipicamente especifica que:

1. Padrões e procedimentos documentados são utilizados na seleção dos subcontratados de software e na gestão dos subcontratos de software.
2. Os acordos contratuais constituem as bases para a gestão de subcontratação.
3. As alterações na subcontratação são realizadas com o envolvimento e a concordância do contratante e do subcontratado.

- Comprom. 2** Um gerente de subcontratação é designado para ser responsável por estabelecer e gerenciar a subcontratação de software.

1. O gerente de subcontratação tem conhecimento e é experiente em engenharia de software ou tem pessoas designadas que possuem esse conhecimento e experiência.
2. O gerente de subcontratação é responsável por coordenar o escopo técnico do trabalho a ser subcontratado e os termos e condições da subcontratação com as partes afetadas.

As equipes de engenharia de sistema e de desenvolvimento de software do projeto definem o escopo técnico do trabalho a ser subcontratado.

As equipes de negócio apropriadas, tais como equipe de compra, financeira e jurídica, estabelecem e monitoram os termos e as condições da subcontratação.

3. O gerente de subcontratação é responsável por:
 - selecionar o subcontratado de software,
 - gerenciar a subcontratação de software,
 - providenciar suporte aos produtos subcontratados.

Habilidades

Habilidade 1

Os recursos e os orçamentos adequados são fornecidos para a seleção do subcontratado de software e para a gestão do subcontrato.

1. Gerentes de software e outras pessoas são designadas para responsabilidades específicas de gestão de subcontratação.
2. São disponibilizadas ferramentas de suporte à gestão de subcontratação.

Exemplos de ferramentas de suporte:

- modelos de estimativas,
- planilhas eletrônicas,
- programas para gerenciamento de projeto e de cronogramas.

Habilidade 2 Os gerentes de software e outras pessoas que estão envolvidas no estabelecimento e na gestão do subcontrato de software são treinados para realizar essas atividades.

Exemplos de treinamento:

- preparação e planejamento para subcontratação de software,
- avaliação da capacidade do processo de software do participante da licitação da subcontratação,
- avaliação das estimativas e planos do participante da licitação da subcontratação,
- seleção de subcontratado,
- gerenciamento do subcontrato.

Habilidade 3 Os gerentes de software e as outras pessoas que estão envolvidas na gestão da subcontratação de software recebem orientação sobre os aspectos técnicos da subcontratação.

Exemplos de orientação:

- domínio da aplicação,
- tecnologias de software que estão sendo aplicadas,
- ferramentas de software que estão sendo usadas,
- metodologias que estão sendo utilizadas,
- padrões que estão sendo utilizados,
- procedimentos que estão sendo utilizados.

Atividades

Atividade 1 O trabalho a ser subcontratado é definido e planejado de acordo com um procedimento formal (documentado).

Esse procedimento tipicamente especifica que:

1. Os produtos e as atividades de software a serem subcontratados são selecionados com base em uma avaliação

equilibrada das características técnicas e não técnicas do projeto.

- As funções ou subsistemas a serem subcontratados são selecionados para casar com as habilidades e capacidades dos potenciais subcontratados.
 - A especificação dos produtos e atividades de software a serem subcontratados são determinadas com base em uma análise sistemática e no particionamento apropriado dos requisitos de software e dos requisitos de sistema.
2. A especificação do trabalho a ser subcontratado e dos padrões e procedimentos a serem seguidos são derivados dos seguintes elementos do projeto:
- contrato,
 - requisitos do sistema alocados ao software,
 - requisitos de software,
 - plano de desenvolvimento de software,
 - padrões e procedimentos de software.
3. O contrato da subcontratação é:
- preparado,
 - revisado,
 - acordado.

Exemplos de pessoas que revisam e acordam o contrato da subcontratação:

- gerente do projeto,
- gerente do projeto de software,
- os gerentes responsáveis pelo software,
- gerente da gestão de configuração de software,
- gerente da garantia da qualidade de software,
- gerente de subcontratação.

- atualizado quando necessário,
- gerenciado e controlado.

“Gerenciado e controlado” implica que a versão do produto em uso em um determinado momento (passado ou presente) é conhecida (ou seja, com controle de versão), e as alterações são incorporadas de forma controlada (ou seja, com controle de alterações).

Se um grau de controle maior que o “gerenciado e controlado” é desejado, então o produto pode ser colocado sob uma forte disciplina de gestão de configuração, como descrito na área-chave de processo *Gestão de Configuração de Software*.

Vide a *Habilidade 1* da área-chave de processo *Planejamento de Projeto de Software* para acessar as práticas que cobrem os conteúdos típicos do contrato.

4. Um plano para a seleção do subcontratado é preparado paralelamente com o contrato da subcontratação e é revisado, quando apropriado.

Atividade 2

O subcontratado de software é selecionado, com base na avaliação da capacidade do participante da licitação de realizar o trabalho, de acordo com um procedimento formal (documentado).

Esse procedimento cobre a avaliação de:

1. Propostas apresentadas para a subcontratação planejada.
2. Registros de desempenho anterior em trabalho similar, se disponível.
3. Localizações geográficas das organizações dos participantes da licitação da subcontratação com relação ao contratante.

Uma gestão efetiva de certas subcontratações pode requerer freqüentes interações face-a-face .

4. Capabilidade de engenharia de software e de gerenciamento de software.

Um exemplo de método para avaliar as capacidades do subcontratado é o método “*SEI Capability Evaluation*” .

5. Quadro de pessoal disponível para realizar o trabalho.

6. Experiência anterior em aplicações similares, incluindo experiência em software da equipe de gestão do subcontratado.
7. Recursos disponíveis.

Exemplos de recursos:

- facilidades,
- hardware,
- software,
- treinamento.

Atividade 3

O acordo contratual entre o contratante e o subcontratado de software é utilizado como base para a gestão de subcontratação.

O acordo contratual documenta:

1. Os termos e as condições.
2. O contrato.

Vide a *Habilidade 1* da área-chave de processo *Planejamento de Projeto de Software* para acessar as práticas que cobrem o conteúdos típicos do contrato.

3. Os requisitos para os produtos a serem desenvolvidos.
4. A lista de dependências entre o subcontratado e o contratante.
5. Os produtos subcontratados a serem entregues ao contratante.

Exemplos de produtos:

- código fonte,
- plano de desenvolvimento de software,
- ambiente de simulação,
- documentação de projeto,
- plano de teste de aceitação.

6. As condições sob as quais as revisões dos produtos serão submetidas.

7. Os procedimentos de aceitação e os critérios de aceitação a serem utilizados na avaliação dos produtos subcontratados antes que eles sejam aceitos pelo contratante.
8. Os procedimentos e critérios de avaliação a serem utilizados pelo contratante para monitorar e avaliar o desempenho do subcontratado.

Atividade 4

O plano de desenvolvimento de software do subcontratado, documentado, é revisado e aprovado pelo contratante.

1. Esse plano de desenvolvimento de software cobre (diretamente ou por referência) os itens apropriados do plano de desenvolvimento de software do contratante.

Em alguns casos, o plano de desenvolvimento de software do contratante pode incluir o plano de desenvolvimento de software do subcontratado, não sendo necessário um plano de desenvolvimento separado para o subcontratado.

Vide a *Atividade 7* da área-chave de processo *Planejamento de Projeto de Software* para acessar as práticas que cobrem o conteúdo do plano de desenvolvimento de software do projeto.

Atividade 5

O plano de desenvolvimento de software do subcontratado, documentado e aprovado, é utilizado para acompanhar as atividades de software e comunicar sua situação.

Atividade 6

As alterações no contrato do subcontratado de software, nos termos e condições da subcontratação e em outros compromissos são decididas de acordo com um procedimento formal (escrito) .

1. Esse procedimento tipicamente especifica que todos os grupos afetados, tanto do contratante como do subcontratado, são envolvidos.

Atividade 7

A gerência do contratante conduz revisões periódicas da situação/coordenação com a gerência do subcontratado de software.

1. Ao subcontratado é fornecida a visibilidade das necessidades e desejos dos clientes e usuários finais do produto, quando apropriado.

Os usuários finais referidos nestas práticas são os usuários finais designados pelo cliente ou representantes dos usuários finais.

2. O desempenho técnico, de custo, de quadro de pessoal e de cronograma do subcontratado é revisado com relação ao plano de desenvolvimento de software do subcontratado.
3. Os requisitos computacionais considerados críticos para o projeto são revisados; as contribuições do subcontratado para as estimativas atuais são acompanhadas e comparadas com as estimativas de cada componente de software como documentado no plano de desenvolvimento de software do subcontratado.
4. As dependências e os compromissos críticos entre o grupo de desenvolvimento de software do subcontratado e outras equipes do subcontratado são tratados.
5. As dependências e os compromissos críticos entre o contratante e o subcontratado são tratados.
 - Os compromissos do subcontratado para com o contratante e os compromissos do contratante para com o subcontratado são revisados.
6. As não-conformidades da subcontratação são tratadas.
7. Os riscos do projeto envolvendo o trabalho do subcontratado são tratados.
8. Os conflitos e questões não solucionáveis internamente pelo subcontratado são tratadas.
9. Os itens de ação são designados, revisados e acompanhados até a sua conclusão.

Atividade 8 **São mantidas revisões e intercâmbios técnicos com o subcontratado de software.**

Essas revisões:

1. Fornecem ao subcontratado a visibilidade das necessidades e dos desejos dos cliente e dos usuários finais, quando apropriado.
2. Monitoram as atividades técnicas do subcontratado.
3. Verificam se a interpretação e a implementação dos requisitos técnicos pelo subcontratado estão em conformidade com os requisitos do contratante.
4. Verificam se os compromissos estão sendo cumpridos.
5. Verificam se as questões técnicas estão sendo resolvidas no momento.

Atividade 9 **As revisões formais para tratar as realizações e os resultados do desenvolvimento de software do subcontratado são conduzidas em marcos selecionados do cronograma de acordo com um procedimento formal (escrito)**

Esse procedimento tipicamente especifica que:

1. As revisões são pré-planejadas e documentadas no contrato.
2. As revisões tratam os compromissos, os planos e a situação das atividades de software do subcontratado.
3. As questões, os itens de ação e as decisões significativas são identificadas e documentadas.
4. Os riscos de software são encaminhados.
5. O plano de desenvolvimento de software do subcontratado é refinado, quando apropriado.

Atividade 10 **A equipe de garantia da qualidade de software do contratante monitora as atividades de garantia da qualidade de software do subcontratado de acordo com um procedimento formal (documentado).**

Esse procedimento tipicamente especifica que:

1. Os planos, os recursos, os procedimentos e os padrões de garantia da qualidade de software do subcontratado são revisados periodicamente para garantir que os mesmos são adequados para monitorar o desempenho do subcontratado.
2. As revisões regulares do subcontratado são conduzidas para garantir que os procedimentos e os padrões aprovados estão sendo seguidos.
 - A equipe de garantia da qualidade de software do contratante verifica as atividades e os produtos de desenvolvimento de software do subcontratado.
 - A equipe de garantia da qualidade de software do contratante audita os registros de garantia da qualidade de software do subcontratado, quando apropriado.
3. Os registros das atividades de garantia da qualidade de software do subcontratado são auditados periodicamente para avaliar até que ponto os planos, os padrões e os procedimentos de garantia da qualidade estão sendo seguidos.

Atividade 11

A equipe de gestão de configuração de software do contratante monitora as atividades de gestão de configuração de software do subcontratado de acordo com um procedimento formal (documentado).

Esse procedimento tipicamente especifica que:

1. Os planos, os recursos, os procedimentos e os padrões de gestão de configuração de software do subcontratado são revisados periodicamente para garantir que eles sejam adequados.
2. O contratante e o contratado coordenam suas atividades em assuntos relativos à gestão de configuração de software para garantir que os produtos do subcontratado possam ser prontamente integrados ou incorporados ao ambiente do projeto do contratante.
3. A biblioteca de configurações base (*baselines*) de software do subcontratado é periodicamente auditada para avaliar até que ponto os padrões e os procedimentos de gestão de

configuração de software estão sendo seguidos e quão efetivos eles são na gestão das configurações básicas de software (*baselines*).

Atividade 12 **O contratante conduz os testes de aceitação como parte da entrega dos produtos de software do subcontratado de acordo com um procedimento formal (documentado).**

Esse procedimento tipicamente especifica que:

1. Os procedimentos e os critérios de aceitação para cada produto são definidos, revisados e aprovados pelo contratante e pelo contratado antes do teste.
2. Os resultados dos testes de aceitação são documentados.
3. Um plano de ação é estabelecido para qualquer produto de software que não passe pelo seu teste de aceitação.

Atividade 13 **O desempenho do subcontratado de software é avaliado periodicamente e a avaliação é revisada com o subcontratado.**

A avaliação do desempenho do subcontratado fornece uma oportunidade para o subcontratado saber se está, ou não, satisfazendo as necessidades de seu cliente (isto é, de seu contratante). Um mecanismo como o de revisões de pagamento de prêmio por desempenho fornece esse tipo de *feedback*, em oposição às revisões técnicas e de coordenação periódica que ocorrem ao longo do projeto. A documentação dessas avaliações também atuam como informações de entrada para as atividades de futuras seleções de subcontratados.

Medição e Análises

Medição 1 **As medições são realizadas e utilizadas para determinar a situação das atividades da gestão de subcontratação de software.**

Exemplos de medições:

- custos das atividades de gestão de subcontratação comparadas com o plano,
- datas reais de entrega dos produtos subcontratados comparadas com o plano,
- datas reais das entregas do contratante para o subcontratado comparadas com o plano.

Verificação da Implementação

Verificação 1

As atividades de gestão de subcontratação de software são revisadas periodicamente com um gerente superior.

A finalidade básica das revisões periódicas pelo gerente superior é fornecer regularmente informações sobre as atividades do processo de software em um nível de abstração adequado. O intervalo de tempo entre as revisões deveria ajustar-se às necessidades da organização, podendo ser prolongado à medida em que estejam disponíveis mecanismos adequados para encaminhamento de exceções.

Vide a *Atividade 1* da área-chave de processo *Acompanhamento e Supervisão de Projeto de Software* para acessar as práticas que cobrem o conteúdo das revisões de supervisão pela gerência superior.

Verificação 2

As atividades de gestão de subcontratação de software são revisadas com o gerente de projeto, periodicamente ou motivado por um evento.

Vide a *Verificação 2* da área-chave de processo *Acompanhamento e Supervisão de Projeto de Software* para acessar as práticas que cobrem o conteúdo típico das revisões de supervisão pela gerência do projeto.

Verificação 3

5. Garantia da Qualidade de Software

Área-chave de processo para o Nível 2: Repetível

O objetivo da Garantia da Qualidade de Software é prover o gerenciamento, com a adequada visibilidade, do processo que está sendo utilizado pelo projeto de software e dos produtos que estão sendo construídos.

A Garantia da Qualidade de Software cobre a revisão e a auditoria de produtos de software e atividades para verificar se os mesmos estão cumprindo os procedimentos e padrões adotados. O objetivo é subsidiar o projeto de software e os gerentes envolvidos com o resultado dessas revisões e auditorias.

A equipe de Garantia da Qualidade de Software trabalha com o projeto de software durante suas fases iniciais para estabelecer planos, padrões e procedimentos que irão agregar valores ao projeto de software e satisfazer as restrições do projeto e as políticas da empresa. Ao participar do estabelecimento de planos, padrões e procedimentos, a equipe de Garantia da Qualidade de Software ajuda a assegurar que os mesmos atendam às necessidades do projeto e verifica se estarão sendo utilizados nas revisões e auditorias ao longo do ciclo de vida do software. A equipe de Garantia da Qualidade de Software revisa as atividades do projeto e propicia o gerenciamento com a visibilidade de quanto o projeto de software está aderente aos planos, padrões e procedimentos estabelecidos.

As questões de não conformidade são primeiramente tratadas internamente ao projeto de software e resolvidas aí mesmo quando possível. As questões que não podem ser resolvidas dentro do projeto são encaminhadas pela equipe de Garantia da Qualidade de Software a um nível adequado de gestão para serem resolvidos.

Esta área-chave de processo cobre as práticas para a equipe realizar a função de Garantia da Qualidade de Software. As práticas que identificam as atividades específicas e os produtos que a equipe de Garantia da Qualidade de Software revisa e/ou audita estão geralmente contidas nas características comuns da Verificação de Implementação de outras áreas-chave de processo.

Metas

- Meta 1** As atividades de Garantia da Qualidade de Software são planejadas.
- Meta 2** A aderência dos produtos de software e das atividades aos padrões, aos procedimentos e aos requisitos estabelecidos é verificada objetivamente.
- Meta 3** As atividades e os resultados das atividades de Garantia da Qualidade de Software são informados às pessoas e aos grupos afetados.
- Meta 4** As questões de não conformidade, que não podem ser resolvidas internamente ao projeto, são levadas ao conhecimento da gerência superior.

Compromissos

- Comprom. 1** O projeto segue uma política organizacional formal (documentada) estabelecida para a implementação de Garantia da Qualidade de Software (GQS).

Esta política tipicamente especifica que:

1. A função de GQS é implementada em todos os projetos de software.
2. A equipe de Garantia da Qualidade de Software tem um canal de comunicação aberto com a gerência superior, que é independente de:
 - gerente de projeto;
 - equipe de desenvolvimento de software do projeto;
 - outras equipes relacionadas, tais como: gestão de configuração e suporte à documentação.

As organizações devem determinar a estrutura organizacional que irá dar suporte às atividades que requerem independência, como a GQS, no contexto de suas metas estratégicas de negócios e ambiente de negócio.

Essa independência deveria:

- proporcionar, às pessoas que estão executando o papel de GQS, a liberdade organizacional necessária para serem “os olhos e os ouvidos” do gerente superior;
- proteger as pessoas que estão executando o papel de GQS da avaliação de desempenho pela gerência do projeto em revisão;
- assegurar ao gerente superior que as informações objetivas sobre os processos e os produtos do projeto de software lhe estão sendo reportadas.

3. O gerente superior revisa periodicamente as atividades e os resultados de GQS.

Habilidades

Habilidade 1

Existe uma equipe responsável pela coordenação e implementação de GQS no projeto.

A equipe pode ser constituída por departamentos, gerentes e pessoas que são responsáveis por um conjunto de tarefas ou atividades. A equipe pode variar de uma a várias pessoas designadas em tempo parcial até várias pessoas trabalhando em tempo integral. A criação da equipe deve levar em conta as tarefas ou atividades atribuídas, tamanho do projeto, estrutura e cultura da organização. Algumas equipes, tais como as de GQS, focam sua atenção nas atividades do projeto, e outras, como as de processos de Engenharia de Software, estão focadas nas atividades de caráter organizacional.

Habilidade 2

Existem recursos e orçamento adequados para a realização das atividades de GQS.

1. É designado um gerente com responsabilidades específicas para as atividades de GQS.
2. É designado um gerente superior, conhecedor do papel da GQS e que tenha autoridade para agir adequadamente, para receber e atuar em questões de não conformidade.
 - Todos os gerentes na cadeia de relato da GQS para o gerente superior são conhecedores do papel, das responsabilidades e da autoridade da GQS.
3. São disponibilizadas ferramentas de suporte às atividades de GQS.

Exemplos de ferramentas de suporte:

- estações de trabalho,
- programas de banco de dados,
- planilhas eletrônicas,
- ferramentas de auditoria.

Habilidade 3

Os membros da equipe de GQS são treinados para poderem realizar suas atividades de GQS.

Exemplos de treinamento:

- habilidades e práticas em engenharia de software,
- papel e responsabilidades da equipe de desenvolvimento e de outras equipes de software,
- padrões, procedimentos e métodos para projeto de software,
- domínio da aplicação do projeto de software,
- objetivos, métodos e procedimentos de GQS,
- envolvimento da equipe de GQS em atividades de desenvolvimento de software,
- uso efetivo de métodos e ferramentas de GQS,
- comunicação interpessoal.

Habilidade 4

Os membros do projeto de software são orientados quanto ao papel, responsabilidades, autoridade e valor da equipe de GQS.

Atividades

Atividade 1 **É elaborado um plano de GQS para o projeto de software, de acordo com um procedimento formal (documentado).**

Esses procedimentos tipicamente especificam que:

1. O plano de GQS é desenvolvido nas fases iniciais e em paralelo com o planejamento geral do projeto.
2. O plano de GQS é revisado pelas pessoas e pelos grupos afetados.

Exemplos de pessoas e grupos afetados:

- gerente de projeto de software,
- outros gerentes de software,
- gerente de projeto,
- representante de GQS do cliente,
- gerente superior a quem a equipe de GQS reporta as questões de não conformidade,
- a equipe de desenvolvimento (incluindo todos os subgrupos, tais como projetistas de software e líderes de implementação).

3. O plano de GQS é gerenciado e controlado.

“Gerenciado e controlado” implica que a versão do produto em uso em um determinado momento (passado ou presente) é conhecida (ou seja, com controle de versão) e as alterações são incorporadas de forma controlada (ou seja, com controle de alterações).

Se um grau de controle maior que o “gerenciado e controlado” é desejado, então o produto pode ser colocado sob a disciplina de Gestão de Configuração, como descrito na área-chave de processo *Gestão de Configuração de Software*.

Atividade 2 **As atividades da equipe de GQS são realizadas de acordo com o plano de GQS.**

O plano cobre:

1. Responsabilidades e autoridade da equipe de GQS.
2. Requisitos de recursos para a equipe de GQS (incluindo pessoal, ferramentas e facilidades).
3. Cronograma e custos das atividades da equipe de GQS do projeto.
4. Participação da equipe de GQS na elaboração do plano de desenvolvimento de software, dos padrões e dos procedimentos para o projeto.
5. Avaliações a serem realizadas pela equipe de GQS.

Exemplos de produtos e atividades a serem avaliados:

- sistema operacional e software de suporte,
- produtos para clientes e produtos internos ao projeto,
- produtos de software e produtos de outros tipos (ex: documentos),
- atividades de desenvolvimento e verificação de produto (ex: execução de casos de teste),
- atividades executadas na criação do produto.

6. Auditorias e revisões a serem conduzidas pela equipe de GQS.
7. Padrões de projeto e procedimentos a serem utilizados como referência para as revisões e auditorias da equipe de GQS.
8. Procedimentos para documentação e acompanhamento de questões de não conformidade até seu encerramento.

Esses procedimentos podem ser incluídos no Plano de GQS ou serem incluídos através de referência a outros documentos.

9. Documentação que a equipe de GQS deve elaborar.
10. Método e frequência de realimentação para a equipe de desenvolvimento de software e para outras equipes relacionadas com as atividades de GQS.

Atividade 3

A equipe de GQS participa da preparação e da revisão do plano de desenvolvimento de software, dos padrões e dos procedimentos.

1. A equipe de GQS provê consulta e revisão dos planos, padrões e procedimentos com relação a:
 - conformidade com a política organizacional,
 - conformidade com padrões e requisitos impostos externamente (ex: padrões requeridos pelo contrato de trabalho);
 - padrões que são apropriados para serem utilizados no projeto,
 - tópicos que deveriam ser tratados no plano de desenvolvimento de software;
 - outras áreas designadas pelo projeto.
2. A equipe de GQS verifica se os planos, padrões e procedimentos estão vigentes e se podem ser utilizados para revisar e fazer a auditoria do projeto de software.

Atividade 4

A equipe de GQS revisa as atividades de desenvolvimento de software para verificar a conformidade.

1. As atividades são avaliadas com relação ao plano de desenvolvimento de software e aos padrões e procedimentos de software adotados.

As práticas que tratam de revisões e auditorias específicas, a serem realizadas pela equipe de GQS, são encontradas nas características gerais da “Verificação da Implementação” em outras áreas-chave de processo.

2. Os desvios são identificados, documentados e acompanhados até o encerramento.
4. As correções são verificadas.

Atividade 5 **A equipe de GQS realiza a auditoria dos produtos de software designados para verificar a conformidade.**

1. Os produtos são avaliados antes de serem entregues aos clientes.
2. Os produtos são avaliados tomando-se como referência os requisitos contratuais e os padrões de software e procedimentos adotados.
3. Os desvios são identificados, documentados e acompanhados até o encerramento.
4. As correções são verificadas.

Atividade 6 **A equipe de GQS reporta periodicamente o resultado de suas atividades para a equipe de desenvolvimento de software.**

Atividade 7 **Os desvios encontrados nas atividades de software e nos produtos são documentados e tratados de acordo com um procedimento formal (documentado).**

Esse procedimento tipicamente especifica que:

1. Os desvios com relação ao plano de desenvolvimento de software, aos padrões definidos para o projeto e aos procedimentos são documentados e resolvidos com os líderes, gerentes de software, ou gerente de projeto sempre que possível.
2. Os desvios com relação ao plano de desenvolvimento de software, aos padrões definidos para o projeto e aos procedimentos, que não puderem ser resolvidos com os líderes, gerentes de software, ou gerente de projeto, são documentados e apresentados a uma gerência superior designada para resolver questões de não conformidade.
3. As questões de não conformidade apresentadas à gerência superior são periodicamente reviradas.

Atividade 8 **A equipe de GQS conduz, de maneira apropriada, as revisões periódicas de suas atividades e de suas decisões com o pessoal de GQS do cliente.**

Medições e Análises

Medição 1 **As medições são feitas e utilizadas para determinar o custo e a situação do cronograma das atividades de GQS.**

Exemplos de medições:

- alcance de marcos das atividades de GQS em comparação com o plano,
- finalização de tarefas, esforços despendidos e orçamento utilizados nas atividades de GQS em comparação com o plano,
- quantidade de auditorias de produtos e revisões de atividades em comparação com o plano.

Verificação da Implementação

Verificação 1 **As atividades de GQS são revisadas periodicamente com um gerente superior.**

A finalidade básica das revisões periódicas pelo gerente superior é fornecer regularmente ciência e compreensão sobre as atividades do processo de software a um nível de abstração adequado. O intervalo de tempo entre as revisões deveria ajustar-se às necessidades da organização, podendo ser prolongado à medida em que mecanismos adequados para encaminhamento de exceções estejam disponíveis.

Vide a *Verificação 1* da área-chave de processo *Acompanhamento e Supervisão de Projeto de Software* para acessar as práticas que cobrem o conteúdo típico da supervisão de longo prazo à 8 0 Tc 4.1022 T

Verificação 2 As atividades de GQS são revisadas pelo coordenador de projeto, tanto periodicamente como motivado por um evento.

Vide a *Verificação 2* da área-chave de processo *Acompanhamento e Supervisão de Projeto de Software* para acessar as práticas que cobrem o conteúdo típico da supervisão de revisões da gerência do projeto.

Verificação 3 Peritos não pertencentes à equipe de GQS revisam periodicamente as atividades e os produtos de software da equipe de GQS do projeto.

6. Gestão de Configuração

Área-chave de processo para o Nível 2: Repetível

O objetivo da Gestão de Configuração de Software é estabelecer e manter a integridade dos produtos do projeto de software ao longo de todo o ciclo de vida de software do projeto.

A Gestão de Configuração de Software compreende a identificação da configuração do software (ou seja, os produtos de software selecionados e suas descrições) em determinado momento, controlando sistematicamente as alterações na configuração e mantendo a integridade e a rastreabilidade da configuração ao longo do ciclo de vida do software. Os produtos colocados sob a gestão de configuração de software incluem os produtos de software que são entregues ao cliente (ex., o documento dos requisitos de software e o código) e os itens que estão identificados com os produtos de software ou necessários para a criação dos mesmos (ex., o compilador).

Uma biblioteca de software contendo as configurações básicas (*baselines*) do software é estabelecida. As alterações nas configurações básicas (*baselines*) e a liberação de produtos de software produzidos a partir da biblioteca de configurações básicas de software são sistematicamente controladas por meio de funções de controle de alterações e de auditoria de configuração da gestão de configuração de software.

Esta área-chave de processo cobre as práticas para a realização da função de gestão de configuração de software. As práticas que identificam itens/unidades de configuração específicos estão contidas nas áreas-chave de processo que descrevem o desenvolvimento e manutenção de cada item/unidade de configuração.

Metas

- Meta 1** As atividades de gestão de configuração de software são planejadas.
- Meta 2** Os produtos de software selecionados são identificados, controlados e disponibilizados.
- Meta 4** As alterações realizadas nos produtos de software identificados são controladas.
- Meta 5** As pessoas e os grupos afetados são informados sobre a situação e o conteúdo das configurações básicas (*baselines*) do software.

Compromissos

- Comprom. 1** O projeto segue uma política formal da organização para a implementação da Gestão de Configuração de Software (GCS).

Esta política tipicamente especifica que:

1. A responsabilidade da GCS de cada projeto é designada explicitamente.
2. A GCS é implementada ao longo de todo o ciclo de vida do projeto.
3. A GCS é implementada para os produtos de software a serem entregues ao cliente, determinados produtos de software internos e determinadas ferramentas de software utilizadas internamente ao projeto (ex., compiladores).
4. Os projetos estabelecem ou têm acesso a um repositório para armazenamento de itens/unidades de configuração e de registros de GCS associados.

Os conteúdos desse repositório são referenciados nestas práticas como “biblioteca da configuração básica (*baseline*) de software”.

As ferramentas e procedimentos utilizados para acessar esse repositório são referenciados nestas práticas como “sistema de biblioteca para gestão de configuração do sistema”.

Os produtos de trabalho colocados sob a gestão de configuração e tratados como entidade isolada são referenciados como itens de configuração.

Os itens de configuração são tipicamente decompostos em componentes de configuração e os componentes de configuração são decompostos em unidades. Em um sistema de hardware/software, todo o software pode ser considerado um item de configuração isolado, ou pode ser decomposto em múltiplos itens de configuração. Nestas práticas, o termo “itens/unidades de configuração” é utilizado para fazer referência aos elementos sob a gestão de configuração.

5. As configurações básicas (*baselines*) do software e as atividades de GCS são revisadas periodicamente.

Habilidades

Habilidade 1

Existe ou é criado um grupo com autoridade para gerenciar a configuração básica (*baseline*) do projeto (isto é, um Conselho de Configuração - CoC).

O CoC:

1. Autoriza a criação das configurações básicas (*baselines*) e a identificação dos itens/unidades de configuração.
2. Representa os interesses do gerente de projeto e de todos os grupos que podem ser afetados pelas alterações nas configurações básicas (*baseline*) do software.

Exemplos de grupos afetados:

- garantia da qualidade de hardware,
- gestão de configuração de hardware,
- engenharia de hardware,
- engenharia de manufatura,
- desenvolvimento de software (incluindo todos os subgrupos, como por ex. projeto de software),
- engenharia de sistema,
- teste de sistema,
- garantia da qualidade de software,
- gestão de configuração de software,
- gestão de contrato,
- suporte à documentação.

3. Revisa e autoriza alterações nas configurações básicas (*baselines*) de software.
4. Autoriza a criação de produtos a partir da biblioteca de configuração básica (*baseline*) de software.

Habilidade 2

Existe uma equipe responsável pela coordenação e implementação da GCS para o projeto.

Uma equipe é uma “reunião” de departamentos, gerentes e pessoas que são responsáveis por um conjunto de tarefas ou atividades. Uma equipe pode variar desde uma única pessoa ou várias pessoas de diferentes departamentos designadas em tempo parcial até várias pessoas trabalhando em tempo integral. A criação da equipe deve levar em conta as tarefas ou atividades designadas, tamanho do projeto, estrutura e cultura da organização. Algumas equipes, tais como as de GQS, focam sua atenção nas atividades do projeto e outras, como as de engenharia de software estão focadas nas macro atividades da organização.

A equipe de GCS coordena ou implementa:

1. A criação e a gestão da biblioteca de configuração básica (*baseline*) do projeto.
2. O desenvolvimento, manutenção e distribuição dos planos, padrões e procedimentos de GQS.

3. A identificação do conjunto de produtos de trabalho a serem colocados sob a GCS.

Um produto de trabalho é qualquer artefato advindo da definição, manutenção ou utilização de um processo de software.

4. O controle do acesso à biblioteca de configuração básica (*baseline*) do projeto.
5. A atualização das configurações básicas (*baselines*) de software.
6. A geração de produtos a partir da biblioteca de configuração básica (*baseline*) do projeto.
7. O registro das ações de GCS.
8. A produção e distribuição dos relatórios de GCS.

Habilidade 3

Os recursos humanos e os materiais adequados são providos para a realização das atividades de GCS.

1. É designado um gerente para assumir as responsabilidades específicas de GCS.
2. São disponibilizadas ferramentas para suportar as atividades de GCS.

Exemplos de ferramentas de suporte:

- workstations,
- sistema de gerenciamento de banco de dados,
- ferramentas para gestão de configuração.

Habilidade 4

Os membros da equipe de GCS são treinados com relação aos objetivos, procedimentos e métodos para realizar suas atividades de GCS.

Exemplos de treinamentos:

- padrões, procedimentos e métodos de GCS,
- ferramentas de GCS.

Habilidade 5 Os membros da equipe de desenvolvimento e outras equipes relacionadas ao software são treinados para realizar suas atividades de GCS.

Exemplos de outras equipes relacionadas ao software:

- garantia da qualidade de software,
- suporte à documentação.

Exemplos de treinamento:

- os padrões, procedimentos e métodos a serem seguidos pelas atividades de GCS realizadas internamente à equipe de desenvolvimento e às outras equipes relacionadas ao software,
- papel, as responsabilidades e a autoridade da equipe de GCS.

Atividades

Atividade 1 É elaborado um plano de GCS para o projeto de software, de acordo com procedimentos documentados

Esses procedimentos tipicamente especificam que:

1. O plano de GCS é desenvolvido nas fases iniciais, ou em paralelo com o planejamento geral do projeto.
2. O plano de GCS é revisado pelos grupos afetados.
3. O plano de GCS é gerenciado e controlado.

“Gerenciado e controlado” implica que a versão do produto em uso em um determinado momento (passado ou presente) é conhecida (ou seja, com controle de versão) e as alterações são incorporadas de forma controlada (ou seja, com controle de alterações).

Se um grau de controle maior que o “gerenciado e controlado” é desejado, então o produto pode ser colocado sob uma forte disciplina de gestão de configuração, como descrito nesta área-chave de processo.

Atividade 2 **Um plano de GCS documentado e aprovado é utilizado como base para a realização das atividades de GCS.**

O plano cobre:

1. As atividades a serem realizadas, o cronograma das atividades, a delegação de responsabilidades e os recursos necessários (incluindo pessoal, ferramentas e facilidades computacionais).
2. Os requisitos e as atividades de GCS a serem realizadas pela equipe de desenvolvimento e por outras equipes relacionadas ao software.

Atividade 3 **Um sistema de biblioteca para gestão de configuração é estabelecido como repositório para as configurações básicas (*baselines*) de software.**

O sistema de biblioteca:

1. Suporta múltiplos níveis de controle de GCS.

Exemplos de situações que conduzem a múltiplos níveis de controle:

- diferenças nos níveis de controle necessários em diferentes estágios do ciclo de vida (ex. maior controle quando o produto atinge um nível maior de maturidade),
- diferenças nos níveis de controle necessários para diversos tipos de sistema (ex. sistemas de software, sistemas que incluem software e hardware etc.)

2. Fornece mecanismos de armazenamento e recuperação de itens/unidades de configuração.
3. Fornece mecanismos de compartilhamento e transferência de itens/unidades de configuração entre os grupos afetados e entre níveis de controle dentro da biblioteca.
4. Auxilia na utilização de padrões para itens/unidades de configuração.
5. Fornece mecanismos de armazenamento e recuperação de versões de arquivos de itens/unidades de configuração.
6. Auxilia a garantir a correta geração de produtos a partir da biblioteca de configuração básica (*baseline*) do software.

7. Fornece mecanismos de armazenamento, atualização e recuperação de registros de GCS.
8. Suporta a produção de relatórios de GCS.
9. Fornece mecanismos de manutenção da estrutura da biblioteca e seu conteúdo.

Exemplos de funções de manutenção de biblioteca:

- backup/recuperação de arquivos,
- recuperação de erros de biblioteca.

Atividade 4

Os produtos de trabalho de software a serem colocados sob gestão de configuração são identificados.

1. Os itens/unidades de configuração são selecionados com base em um critério documentado.

Exemplos de produtos de trabalho de software que podem ser identificados como itens/unidades de configuração:

- documentos de processo (ex. planos, padrões/procedimentos),
- requisitos de software,
- projeto de software,
- unidades de código de software,
- procedimentos de testes de software,
- sistemas construídos para a realização da atividade de teste,
- sistemas construídos para serem entregues aos clientes ou usuários finais,
- compiladores,
- outras ferramentas de suporte.

2. São atribuídos identificadores únicos aos itens/unidades de configuração.
3. As características de cada item/unidade são especificadas.
4. A configuração básica (*baseline*) de software de cada item/unidade é especificada.
5. O ponto do desenvolvimento em que cada item/unidade de configuração será colocado sob a gestão de configuração é especificado.

6. A pessoa responsável por cada item/unidade de configuração (isto é, o proprietário, do ponto de vista de gestão de configuração) é identificada.

Atividade 5 As solicitações de alterações e relatórios de problemas para todos os itens/unidades de configuração são iniciados, revisados, aprovados e encaminhados de acordo com um procedimento documentado.

Atividade 6 As alterações das configurações básicas (*baselines*) são controladas de acordo com um procedimento documentado.

Esse procedimento tipicamente especifica que:

1. São realizadas revisões e/ou testes de regressão para garantir que as alterações não causaram efeitos indesejáveis na configuração básica (*baseline*).
2. Somente os itens/unidades que foram aprovados pelo CoC são incluídos na biblioteca de configuração básica (*baseline*) do software.
3. Os itens/unidades de configuração são inseridos e retirados de forma a manter a correção e a integridade da biblioteca de configuração básica (*baseline*) do software.

Exemplos de passos da inserção/retirada:

- verificar se as revisões estão autorizadas,
- criar um log de alterações,
- manter uma cópia das alterações,
- atualizar a biblioteca de configuração básica (*baseline*) do software,
- arquivar a configuração básica (*baseline*) do software que foi substituída.

Atividade 7 Os produtos são criados a partir da biblioteca de configuração básica do software (*baseline*) e suas versões são controladas de acordo com um procedimento documentado.

Esse procedimento tipicamente especifica que:

1. O CoC autoriza a criação de produtos a partir da biblioteca de configuração básica (*baseline*) do software.
2. Os produtos gerados a partir da biblioteca de configuração básica (*baseline*) do software, tanto para uso interno quanto externo, só são construídos a partir dos itens/unidades de configuração existentes na biblioteca.

Atividade 8

A situação dos itens/unidades de configuração é registrada de acordo com um procedimento documentado.

Esse procedimento tipicamente especifica que:

1. As ações de gestão de configuração são registradas com detalhes suficientes de tal forma que o conteúdo e a situação de cada item/unidade de configuração sejam conhecidos e que as versões anteriores possam ser recuperadas.
2. A situação corrente e o histórico (isto é, alterações e outras ações) de cada item/unidade de configuração são mantidos.

Atividade 9

Os relatórios padrão que documentam as atividades da GCS e o conteúdo da configuração básica (*baseline*) do software são desenvolvidos e disponibilizados para as pessoas e para os grupos afetados.

Exemplos de relatórios:

- notas de reunião do CoC,
- resumo das solicitações e respectivas situações,
- resumo dos relatórios de problemas e respectivas situações (incluindo correções),
- resumo das alterações realizadas na configuração básica (*baseline*) do software,
- histórico de revisões dos itens/unidades de configuração,
- situação da configuração básica (*baseline*) do software,
- resultados das auditorias da configuração básica (*baseline*) do software.

Atividade 10 **As auditorias na configuração básica (*baseline*) do software são conduzidas de acordo com um procedimento documentado.**

Esse procedimento tipicamente especifica que:

1. Existe uma preparação adequada para as auditorias.
2. A integridade da configuração básica (*baseline*) do software é avaliada.
3. A estrutura e facilidades do sistema de biblioteca de gestão de configuração são revisadas.
4. A completude e a correção do conteúdo da biblioteca de configuração básica (*baseline*) do software são verificadas.
5. A conformidade com os padrões e procedimentos aplicáveis da gestão de configuração de software é verificada.
6. Os resultados das auditorias são reportados ao gerente de projeto de software.
7. As ações decorrentes das auditorias são acompanhadas até sua efetiva conclusão.

Medições e Análises

Medição 1 **As medições são feitas e utilizadas para determinar a situação das atividades da Gestão de Configuração de Software.**

Exemplos de medições:

- número de solicitações de alteração por unidade de tempo,
- alcance de marcos das atividades de GCS em relação ao plano,
- trabalhos concluídos, esforços e orçamentos despendidos nas atividades de GCS.

Verificação da Implementação

Verificação 1 **As atividades de Gestão de Configuração de Software são revisadas periodicamente pelo gerente superior.**

A finalidade básica das revisões periódicas pelo gerente superior é fornecer regularmente ciência e compreensão sobre as atividades do processo de software a um nível de abstração adequado. O intervalo de tempo entre as revisões deveria ajustar-se às necessidades da organização, podendo ser prolongado à medida em que mecanismos adequados para encaminhamento de exceções estejam disponíveis.

Vide a *Verificação 1* da área-chave de processo *Acompanhamento e Supervisão de Projeto de Software* para acessar as práticas que cobrem o conteúdo típico da supervisão de revisões da gerência superior.

Verificação 2 **As atividades de Gestão de Configuração de Software são revisadas com o gerente de projeto, tanto periodicamente como motivado por um evento.**

Vide a *Verificação 2* da área-chave de processo *Acompanhamento e Supervisão de Projeto de Software* para acessar as práticas que cobrem o conteúdo típico da supervisão de revisões da gestão do projeto.

Verificação 3 **A equipe de Gestão de Configuração periodicamente faz a auditoria das configurações básicas (*baselines*) para verificar se elas estão de acordo com a documentação que as define.**

Verificação 4 **A equipe de Garantia da Qualidade de Software revisa e/ou a faz a auditoria das atividades e produtos da GCS e reporta os resultados.**

Vide a área-chave de processo Garantia da Qualidade de Software.

No mínimo essas revisões e auditorias verificam:

1. A conformidade aos padrões e procedimentos da GCS, pela:
 - equipe de GCS,
 - CoC,
 - equipe de desenvolvimento,
 - outras equipes de software relacionadas.

2. A ocorrência de auditorias periódicas na configuração básica (*baseline*) do software.