

GCO: Por onde começar?

Como implementar a gerência de configuração a partir do zero



Samuel Diniz Casimiro

samuel.casimiro@gmail.com

É servidor público e trabalha como analista de sistemas em um grande órgão do governo; trabalhou cerca de 10 anos na área de desenvolvimento de sistemas de um grande banco brasileiro; é certificado SCJP e SCWCD; especialista em soluções J2EE para intranets; consultor de TI para pequenas e médias empresas; e crítico ferrenho dos modelos de desenvolvimento adotados atualmente nas grandes empresas.

Imagine que, de repente, o seu gerente entra na sala e te diz: “A Diretoria decidiu que vai implementar o MPS.BR aqui na nossa empresa. Vamos montar uma equipe e você foi um dos escolhidos. Resolvemos que você ficará encarregado do processo Gerência de Configuração (ler **Nota 1**).”

De que se trata o artigo?

Veremos como se pode começar a implementação da Gerência de Configuração com o objetivo de obter os níveis G e F do MPS.BR. O escopo deste artigo é, portanto, a gerência de configuração de software, focada no ciclo de desenvolvimento – e não a gerência de configuração como abordada pelo ITIL, que é mais abrangente.

Para que serve?

O propósito do processo Gerência de Configuração – GCO – é registrar e garantir a integridade dos produtos de trabalho dos demais processos relacionados com o ciclo de desenvolvimento de software. Sem o processo GCO, o software evolui sem controle e isso pode gerar um caos nas bibliotecas de software e na documentação de projetos e processos de forma geral.

Em que situação o tema é útil?

O processo GCO, como descrito no MPS.BR, descreve uma série de boas práticas para a implementação de um Sistema de Gerência da Configuração e de atividades que vão garantir um melhor controle dos produtos de trabalho dos processos relacionados com o desenvolvimento de software.



Nota do DevMan 1

Gerência de Configuração

Sob a perspectiva de desenvolvimento, a Gerência de Configuração de Software abrange três sistemas principais: controle de modificações, controle de versões e controle de gerenciamento de construção.

O sistema de controle de versões permite que os artefatos sob Gerência de Configuração evoluam de forma distribuída, concorrente e disciplinada, evitando perdas ou sobreposições durante o desenvolvimento e a manutenção do artefato. Podemos citar como exemplos de ferramentas de mercado: CVS, Subversion, IBM Rational ClearCase e Microsoft Visual Source Safe.

Um item, ao ser desenvolvido, evolui até que atinja um estado em que atenda aos propósitos para o qual foi criado. Isso implica em diversas alterações, gerando uma versão do item a cada estado (Munch, 1996). Para estabelecer o controle sobre as diversas versões, todas as versões devem ser armazenadas e identificadas. Isso, geralmente, é feito com o auxílio de uma ferramenta.

A versão do item pode ser incluída no esquema de identificação ou ser acessível a partir de uma tabela à parte. É conveniente que o esquema de identificação das versões dos itens seja feito em forma de árvore, pois ao mesmo tempo em que mantém um histórico das versões dos itens, permite identificação única e ramificações a partir de qualquer versão.

O sistema de controle de modificações armazena todas as informações geradas durante o andamento das solicitações de modificação e relata essas informações aos participantes interessados e autorizados. Podemos citar como exemplos de ferramentas de mercado: Bugzilla, Jira, Trac e IBM Rational ClearQuest.

O objetivo dessa tarefa de gerência de configuração é relatar a todas as pessoas envolvidas no desenvolvimento e na manutenção do software as seguintes informações sobre as alterações na configuração de software:

O que aconteceu?

Quem o fez?

Quando aconteceu?

O que mais será afetado?

Para isso, deve ser criado um banco de dados sobre as ocorrências na gerência de configuração. Esse banco de dados deve estar disponível aos desenvolvedores com acesso através de palavras-chave. Além disso, deve ser gerado regularmente um relatório de situação para informar as alterações mais importantes. O acesso rápido às informações sobre a configuração agiliza o processo de desenvolvimento e melhora a comunicação entre as pessoas, o que é uma maneira de eliminar muitos problemas relativos à modificação do mesmo item de informação, com intenções diferentes e conflitantes.

O sistema de gerenciamento de construção automatiza o processo de transformação dos diversos artefatos do software que compõem um projeto em um sistema executável propriamente dito. Este processo é nomeado construção do software que, por exemplo, testa e empacota a aplicação java como um arquivo jar. Este processo ocorre de forma aderente às normas, procedimentos, políticas e padrões definidos para o projeto. Podemos citar como exemplos de ferramentas de mercado: Maven e Apache Ant.

As vantagens da utilização da Gerência de Configuração de Software são inúmeras. Dentre elas, podemos listar: (1) ganho de produtividade e eficiência; (2) diminuição do retrabalho e dos erros; (3) aumento da disciplina no processo de desenvolvimento; (4) aumento da memória organizacional; (5) acesso às informações qualitativas e quantitativas referentes ao processo de desenvolvimento, como por exemplo, medida de esforço para efetuar uma alteração e frequência de modificações por componente; (6) possibilidade de estabelecer uma trilha de auditoria indicando por que, quando e por quem um artefato foi alterado; (7) auxílio à gerência de projetos e (8) garantia de ambiente estável no qual o produto deve ser desenvolvido.

Qualquer semelhança com a realidade é mera coincidência. Também, quem mandou você ficar enviando e-mails para os gerentes com sugestões sobre melhoria de processos? Agora eles imaginam que você é alguém com perfil para trabalhar na implementação do MPS.BR. E mais, assumiu um dos maiores desafios: a gerência de configuração.

Mas não precisa se desesperar. Afinal, você se considera muito inteligente, não é verdade? Então vai dar conta do recado (e nós da Engenharia de Software estamos aqui para te ajudar, não é verdade?). Passado o susto dessa notícia, é bem provável que a sua empresa providencie algum tipo de treinamento para você. Além disso, a essa altura do campeonato, você provavelmente já sabe o que é gerência de configuração (confira o artigo “Gerência de Configuração de Software”, da 2ª. Edição da revista Engenharia de Software). Mais daí surge a pergunta: Por onde começar?

Bom, você que leu os Guias de Implementação da Softex (veja os links no final do artigo), já percebeu que o processo Gerência de Configuração – GCO – não é exigido no primeiro Nível do modelo, o Nível G. A GCO só é exigida do Nível F em diante. Mas não fique excessivamente tranqüilo. É um erro achar que você só deve se preocupar com GCO depois que a sua empresa obter o Nível G. Na verdade, esse é um dos maiores erros que levam as

empresas a sofrerem tanto no momento de implementar a GCO. Como melhores práticas, dispare ações de GCO imediatamente, pois o tempo corre contra você. Há muito trabalho a se fazer antes de seus colegas responsáveis pela Gerência de Projetos – GPR – e pela Gerência de Requisitos – GRE – obterem o Nível G. A GCO precisa estar lado a lado dos demais processos, desde o seu início. Por quê? Porque o processo GCO se cruza com todos os demais processos, inclusive com ele mesmo, por meio do atributo de processo RAP 13: “Os produtos de trabalho são colocados em níveis apropriados de controle”. Isso é gerência de configuração. Por isso, esteja preparado desde o início. É mais fácil começar certo do que arrumar depois. Como diz o ditado, é melhor prevenir do que remediar!

E vamos logo ao trabalho! Veremos a seguir uma série de pontos importantes por onde você pode começar, focando apenas na GCO. Este artigo não tem a pretensão de avaliar cada cruzamento da GCO com todos os demais processos. Quando falarmos de cruzamentos, vamos nos concentrar nos processos dos níveis G e F, que é por onde você vai começar e é quando a GCO precisa ser consolidada na organização. Além do mais, a curva de aprendizado é muito mais alta no início. Começando certo fica mais fácil de implementar a GCO nos próximos níveis.

No mais, vale ressaltar que, ao decidir implementar o MPS.BR, sua empresa provavelmente já tomou uma série de providências. Coisas tais como iniciar um projeto, definir patrocinadores, alocar pessoas como você, providenciar treinamentos, talvez contratar uma consultoria, definir um cronograma, designar responsabilidades e garantir a colaboração de todos na empresa. Não é o objetivo deste artigo tratar desses assuntos, que estão muito mais relacionados com gestão do que com a GCO propriamente dita. Vamos imaginar que os gerentes da sua empresa fizeram seu dever de casa. Em outros artigos, talvez possamos nos concentrar melhor no papel dos gerentes com relação à implementação desse modelo de melhoria. Aqui, por enquanto, vamos nos preocupar nos aspectos práticos e operacionais, ok?

Comece pelos RAPs

Como você já deve saber, os resultados esperados exigidos pelo modelo se dividem em Resultados dos Atributos de Processo – RAPs – e em Resultados Esperados do Processo – que aqui vamos chamar simplesmente de “resultados esperados”. Ao passo que os RAPs se referem a atributos genéricos que se aplicam a todos os processos e indicam a capacidade deles, os resultados esperados são específicos de cada processo.

Se você já teve a chance de acompanhar uma avaliação MPS.BR, ou se pelo menos já participou do preenchimento das planilhas de indicadores (também chamada “planilha de evidências”), então deve ter reparado que os RAPs são avaliados por último, depois dos resultados esperados de cada processo. Nos relatórios de GAP-Analysis (para mais detalhes, veja o **quadro** “Avaliação preliminar: Vale a pena?”), os RAPs também ficam por último. E o mesmo ocorre nos guias de implementação de cada nível. Até o próprio RAP 1, “O processo atinge seus resultados esperados”, coloca os RAPs em segundo plano. O que se pretende dizer com isso é: se você não tomar cuidado, vai acabar caindo na armadilha de deixar a implementação dos RAPs para depois dos resultados esperados. E isso, prezado leitor, é um grande erro! Como assim? Vamos esclarecer melhor essa questão.

Os RAPs, especialmente do 2 ao 10 e do 15 em diante, fornecem a base para a implementação de todo o processo. Estamos falando da institucionalização do processo, ou seja, o seu alicerce. Por exemplo, o RAP 2 fala de uma “política organizacional” para o processo. Pergunta: como você vai implementar a operacionalização de um processo sem uma política organizacional estabelecida para ele? Outro exemplo, como você vai definir os pontos em que os resultados esperados vão ser produzidos se o processo ainda não foi planejado, conforme exige o RAP 3?

É claro que nem todos os RAPs devem ser observados antes dos resultados esperados, mas provavelmente você vai ter mais facilidade se começar por eles, especialmente pelo RAP 2. Uma boa dica é: dê uma estudada geral nos RAPs, implemente o RAP 2 para GCO e depois comece a implementar os resultados esperados um a um sempre repassando os demais RAPs e verificando o que se aplica ao momento. Vamos falar mais sobre isso a seguir.

Avaliação preliminar: Vale a pena?

Uma prática que se tem visto em algumas empresas, especialmente durante a implementação dos níveis G e F, é a contratação de uma avaliação preliminar – também chamada de GAP-Analysis (análise das lacunas). Essa avaliação também é feita por uma instituição avaliadora autorizada pela Softex e consiste em verificar se a empresa apresenta alguma lacuna com relação a um nível de maturidade previamente escolhido pela empresa. Assim como a avaliação oficial, essa avaliação preliminar gera, ao final da análise, um relatório que aponta os resultados esperados, gerais ou específicos, para os quais não foram encontradas evidências. Por isso, não espere algo parecido com uma consultoria. Uma consultoria de implementação vai indicar não só as lacunas, mas também *meios* de sanar tais lacunas. O máximo que você pode conseguir de uma avaliação preliminar é uma breve reunião com os avaliadores para tirar algumas dúvidas sobre o relatório.

Por isso, devido ao fato de ser tão precária e não indicar os *meios* de se sanar as lacunas, alguns especialistas acreditam que essa avaliação preliminar seja totalmente dispensável. Assim, se a sua empresa não está disposta a contratar uma consultoria de implementação, sugiro também não contratar essas avaliações preliminares. Talvez seja mais interessante contratar um profissional de mercado com boa experiência em avaliações. Um profissional assim, dentro da sua empresa, vai não só apontar as lacunas, como também propor meios para evidenciar todos os resultados que se esperam.

Primeiro a política

A política organizacional é de suma importância para que você saiba o que a alta administração da sua empresa espera do processo GCO. Tudo bem se você fizer parte do *staff*: um analista ou talvez um técnico. Mas, se você está conduzindo e coordenando a implementação do processo GCO, é sua responsabilidade provocar a alta administração para construir essa política e se comprometer com ela. Só não pense que eles vão fazer todo o trabalho para você. Na prática, provavelmente você terá que construir essa política, propô-la, e eles vão apenas analisá-la e aprová-la. É importante lembrar que essa aprovação tem que ser registrada, pois isso é exigido pelo MPS.BR. Essa aprovação pode ser uma ata de reunião, um comunicado da alta administração para os seus funcionários divulgando a política, etc.

Caso você esteja tendo dificuldades em construir uma política de GCO, pode-se fazer um *benchmarking* com outras empresas. Mantenha contatos com pessoas da área de engenharia de software. Esses contatos podem te ajudar com informações e outras dicas. Uma teia de relacionamentos assim constitui um excelente fórum para discutir assuntos relacionados com MPS.BR. Sites de relacionamento, como o LinkedIn, Facebook e Orkut são ótimas ferramentas para isso. Seja membro das comunidades mais populares que tratam de MPS.BR e CMMI. Existem grupos de discussão muito pertinentes também no Yahoo! Groups e no Google Groups. Outra fonte de informação para a sua política de GCO são as demais políticas que porventura existam na sua organização, como a política de segurança da informação. Aproveite a estrutura e o formato que a sua organização utiliza para essas políticas – é um bom começo.

Depois o plano

Feita a política, é hora de passarmos para o RAP 3. Esse RAP fala do plano do processo. O plano nada mais é do que um documento que descreve o processo e define como ele deve ser executado, ou seja, estamos falando de planejamento. Isso envolve desenhar o processo, suas atividades e seus produtos ou artefatos. O plano também deve especificar os recursos que serão utilizados na execução do processo, as pessoas e os papéis que desempenharão as atividades (RAP 5 e 6). Por fim, o plano do processo deve deixar claro o momento em que as atividades serão executadas. Por tudo isso, muitos consideram o plano como o artefato mais importante do processo.

Ainda no plano e já pensando no RAP 4, aproveite para definir atividades de monitoramento do processo, ou seja, formas de verificar se o processo está sendo executado conforme as especificações do plano. São as famosas atividades de verificação. Pode-se aproveitar o plano também para atender ao RAP 12. Esse RAP, quando aplicado ao processo GCO, trata da gerência de configuração do próprio processo. Se você já definiu os produtos de trabalhos das atividades do processo, agora é o momento de definir os níveis de controle de cada produto ou artefato. A definição dos níveis de controle responde às seguintes perguntas: O artefato será armazenado em meio controlado? Onde? Quem poderá acessar? Alterações nesse artefato precisam ser controladas? Como? Será utilizado um sistema de versionamento? Esse artefato ficará sob baseline? Essas perguntas podem ser respondidas no próprio plano do processo de GCO. O MPS.BR sugere vários níveis de controle (para uma descrição de alguns deles, veja a **Tabela 1** – Níveis de controle). Mas nada impede que você defina outros níveis que melhor se adaptem à realidade da sua organização.

Treinamento não é tempo perdido

Outra coisa importante que não se deve deixar para depois é o treinamento das pessoas (RAP 7). Comece solicitando treinamento para você mesmo, caso ainda não tenha feito.

Em seguida, promova treinamento para aqueles que foram indicados pela administração para executar as atividades do processo definidas no plano de GCO. Nessa fase, pode ser de ajuda encorajar que algumas pessoas se certifiquem. Apesar de as certificações não terem muito valor prático, pois elas auferem basicamente apenas conhecimento teórico, a busca por elas acaba gerando alguma empolgação das pessoas pelo assunto da certificação. E empolgação é muito importante nessa fase da implementação dos processos. Precisamos de pessoas engajadas, dispostas a promover a mudança organizacional necessária para implementar o processo GCO. Precisamos de evangelistas. Quanto mais gente do seu lado, melhor.

Reparou como tanta coisa depende dos RAPs 2 e 3? Vamos agora passar para outro aspecto muito relacionado com os RAPs 12 e 13.

Preparando processos e artefatos

Os RAPs 12 e 13 estão diretamente relacionados com GCO. O próprio guia de implementação deixa isso bem claro. O RAP 12 exige que sejam definidos requisitos de GCO para todos os produtos de trabalho, isto é, para todos os artefatos consumidos e produzidos nas atividades dos demais processos. Lembra quando falamos do plano de GCO, que teríamos que definir níveis de controle para os produtos de trabalho do próprio processo GCO? Pois é. Vamos precisar fazer isso para os demais processos também. Em que plano você vai fazer isso? No plano de GCO? Ou nos planos de cada processo? Bom, essas são questões que você terá que responder. Uma sugestão: defina isso em um único catálogo para todos os processos, inclusive para o próprio processo GCO (vamos falar mais sobre esse catálogo abaixo).

É nos RAPs 12 e 13 que a GCO se cruza com os demais processos. Daí a importância de já nos preocuparmos com isso desde o início da implementação dos demais processos, mesmo que você esteja ainda no Nível G.

Nível de controle	Descrição
Observação: Os níveis de controle descritos abaixo podem ser combinados conforme a necessidade.	
Controle de acesso por autenticação	Esse é, talvez, o nível de controle mais precário. Nesse nível, o sistema não faz qualquer tipo de restrição ao acesso do recurso – apenas identifica o usuário mediante autenticação, geralmente gravando um log de acesso.
Controle de acesso por autorização	Esse nível de controle é geralmente implementado em conjunto com a autenticação e consiste em liberar o acesso do usuário apenas a recursos previamente autorizados. Essa autorização geralmente é concedida por usuários de níveis hierárquicos superiores.
Controle pessimista	O procedimento de check-out de um recurso impede que ele seja acessado por outro usuário até que seja realizado o procedimento de check-in. Em outras palavras, as alterações só podem ser feitas por um usuário de cada vez.
Controle otimista	O procedimento de check-out não impede que ele seja acessado por outro usuário. Entretanto, ao realizar o procedimento de check-in, o sistema é capaz de verificar conflitos nas alterações e oferecer sugestões para resolvê-los. Esse nível de controle é muito utilizado e está disponível em muitas ferramentas de controle de versões.
Apenas armazenar	Envolve apenas armazenar o recurso em local pré-estabelecido. Por exemplo, em uma pasta na rede local ou uma pasta de um repositório do controle de versões. O local de armazenamento, geralmente é descrito no plano do processo ou em um catálogo de artefatos, se houver.
Colocar sob baseline	Envolve armazenar o recurso em um repositório do controle de versões e definir procedimentos para proteger a baseline. Isso geralmente envolve o trabalho com branches e a utilização de recursos de rotulação de versões (tags). A colocação de itens sob baseline geralmente envolve um processo formal de verificação da qualidade e aprovação das alterações.

Tabela 1. Níveis de controle

Defina logo cedo os dados de controle

Mesmo que a sua empresa ainda esteja implementando o Nível G, é interessante já começar definindo os níveis de controle para os artefatos do processo GRE e GPR. E, para definir os níveis de controle, a documentação dos processos e dos artefatos precisa estar preparada para isso. Quando vocês estiverem no Nível F, bastará observar os níveis de controle que vocês já definiram anteriormente.

Para começar, toda a documentação dos processos e os templates dos artefatos precisam conter dados suficientes para as atividades de controle de versões e controle de modificações. A GCO é a base para o gerenciamento de mudanças.

Convém que os documentos do processo tenham um número de versão e que a documentação do processo indique claramente o período de vigência de cada versão do processo e quais versões dos templates fazem parte daquela versão do processo. Isso é importante para que aqueles que vão executar o processo saibam exatamente quais versões da documentação e dos templates devem ser utilizadas em um dado momento.

Além do controle de versões, a GCO trata também do controle de modificações. Assim, é importante identificar na documentação do processo e nos templates não só um número de versão, mas também quem foi responsável por aquela versão, quando ela passou a vigorar e o quê motivou cada alteração. No futuro, quando você estiver implementando o Nível F, vai utilizar esses dados para realizar o controle de alterações nos artefatos. Você deve ter informação suficiente para responder às seguintes perguntas: Que versão do processo ou template devo utilizar? Qual a diferença básica entre a versão atual e as anteriores?

Catálogo de artefatos

Entretanto, não basta identificar a versão dos artefatos e registrar aquilo que motivou cada modificação. Para que um sistema de GCO se torne viável, é necessário definir padrões para identificar cada artefato de forma inequívoca. Em outras palavras, você precisa definir padrões de nomenclatura e um documento que descreva de forma simples, porém suficiente, o propósito de cada artefato. Para isso, a nossa sugestão é que você desenvolva uma espécie de sistema catálogo que serviria de repositório para toda a documentação dos processos e os seus respectivos templates de artefatos. Esse catálogo implementaria mecanismos de classificação dos documentos, ferramentas de pesquisa e controle de versões e de modificações. Um catálogo assim poderia ser implementado com base em alguma ferramenta para controle de versões, como o CVS. Esse catálogo seria parte integrante do sistema de GCO. Vamos falar mais sobre isso depois do próximo tópico.

Resultados esperados mais críticos

Vamos agora supor que você já implementou o Nível G e que os processos GPR e GRE estão preparados para implementar definitivamente os RAPs 12 e 13, que tratam de GCO, conforme sugerido acima. O próximo passo agora é implementar o Nível F e o processo GCO propriamente dito. O que fazer agora?

O que é baseline?

Este é um dos conceitos mais importantes da GCO. Se formos traduzir ao pé da letra, baseline significa literalmente “linha base”. De início, isso pode parecer estranho, mas a tradução utilizada nos textos em português sobre ITIL talvez possa ajudar: “Configuração de referência”. Essa definição representa melhor o real significado do termo.

Baseline é justamente isso, a configuração de referência de um ou mais ICs. E, nesse ponto, a configuração se traduz em versão. Em outras palavras, uma baseline representa um ou mais ICs cuja versão sirva de referência para o processo de desenvolvimento de software. Vamos exemplificar.

Imagine que a sua empresa possua três ambientes: desenvolvimento, testes/homologação e produção. Concorde que as versões que estão rodando em produção servem de referência para futuras alterações? Os ICs de produção, em geral, compõem o que chamamos de “baseline de sistema”. Se, por exemplo, alguma implantação apresentar problemas, podemos restaurar a última baseline de sistema que, teoricamente, tudo voltará ao normal. A baseline de sistema serve de base também para futuros projetos. Aplicações cliente/servidor, em geral, possuem apenas uma baseline de sistema. Já aplicações desktop podem possuir várias (ex.: Word 97, Word 2000, Word XP).

Vale a pena também falar de outro tipo de baseline, a “baseline de projeto”. Digamos que você esteja desenvolvendo um software de forma iterativa em um projeto. A cada iteração, novos ICs são desenvolvidos, testados e homologados. De forma que, ao final de uma iteração, você terá um conjunto de ICs mais ou menos estáveis, que foram testados e aprovados. Concorde que esses ICs servirão de base para a próxima iteração do projeto? É justamente isso que chamamos de baseline de projeto: Um conjunto de ICs cuja configuração (leia-se “versão”) servirá de base para a continuação do projeto.

Portanto, baseline é um conjunto de ICs em versões de referência, isto é, que servem de base para outras versões, novos ICs, etc. E, em se tratando de versões de referência, elas devem ser controladas com rigor. Pressupõe-se que ICs que compõem uma baseline foram de alguma forma testados e aprovados para se garantir certo grau de confiança nessa baseline.

Vamos repassar alguns resultados esperados mais críticos e ver como, na prática, poderemos alcançar esses resultados e conseguir as evidências necessárias à certificação.

Sistema de GCO

Sem dúvida, o resultado esperado mais crítico é o primeiro, GCO 1. Afinal, se um sistema de gerência da configuração (SGCO) for bem implementado, vários resultados esperados – e até alguns RAPs – serão obtidos como consequência disso. O problema é que não existe ainda um único sistema que implemente plenamente a GCO como o MPS.BR exige. O que temos são alguns sistemas que implementam subconjuntos das funções do GCO.

Como já foi explicado em artigos anteriores (confira o artigo “Gerência de Configuração de Software”, da 2ª. Edição), o sistema de GCO, como descrito no MPS.BR, deve contemplar três subconjuntos principais de funções: controle de versões, controle de modificações e controle de construção ou liberação. Mas, como acabamos de dizer, não existe ainda um único

sistema que ofereça todas essas funções num único pacote. O que existe são alguns sistemas para cada um desses conjuntos de funções, que podem ou não ser integrados.

Por exemplo, para o controle de versões, temos o conhecido CVS, o Subversion, o IBM Rational ClearCase e o Microsoft Visual Source Safe (VSS), dentre outros. Para o controle de modificações, temos o Bugzilla, Jira, Trac e o IBM Rational ClearQuest, só para citar alguns. Para o controle de construção, a oferta é menor, pois esse tipo de ferramenta depende muito da sua infraestrutura de TI e, em geral, é desenvolvido em cada empresa. É claro que quando o assunto é construção (build), não podemos deixar de lado o Apache Ant. Contudo, cabe à empresa desenvolver seus próprios scripts de build e, assim, automatizar as atividades de liberação de software. O difícil é integrar tudo isso para garantir a rastreabilidade.

Diante dessa dificuldade, o quê fazer? Bom, se você trabalha em uma grande empresa, sugiro que você avalie a aquisição de ferramentas proprietárias, como as da IBM. As ferramentas já citadas se integram perfeitamente e podem, juntas, fornecer todas as funções de um sistema de GCO. Entretanto, se você trabalha em uma pequena empresa, adquirir uma ferramenta assim talvez não seja viável. Nesse caso, sugiro que você desenvolva seu próprio sistema de GCO, provavelmente utilizando ferramentas opensource como o CVS e o Apache Ant para o controle de versões e liberações, respectivamente, e implemente as funções de controle de modificações por conta própria, baseado em banco de dados mesmo e alguma linguagem de programação da sua preferência. Se a sua empresa utiliza o Eclipse como IDE, existem alguns plugins muito interessantes que podem ser integrados ao CVS e ao Apache Ant como o Synergy ALM, TD/OMS e o Aldon Application Lifetime Management Suite.

Identificação dos itens de configuração

Vamos agora ao GCO 2, que em minha opinião também é crítico (afinal, o que NÃO é crítico?). Esse resultado esperado trata da identificação dos itens de configuração. Como o propósito deste artigo é falar da implementação de GCO, não vamos nos deter aqui na definição do que é um item de configuração – IC. Se você solicitou o treinamento sugerido no início deste artigo, já deve saber muito bem do que se trata. Para o que pretendemos aqui, vamos considerar como itens de configuração todos os artefatos críticos para os projetos de desenvolvimento de software. O MPS.BR exige que esses ICs sejam identificados (elencados) com base em critérios e que os níveis de controle sejam definidos.

Bom, já falamos sobre níveis de controle. Mas o que vem a ser esses critérios de identificação? É tudo

aquilo que o levou a tratar um artefato como IC. Em geral, isso é feito com base na criticidade do artefato. Por exemplo, a especificação do software é algo crítico, não concorda? Modificações na especificação geram impactos em todo o processo de desenvolvimento. Portanto, sem dúvida, a especificação deve ser encarada como um IC. Casos de uso, diagramas, modelos de dados e o próprio código fonte também devem ser tratados como ICs, só para citar alguns produtos de trabalho críticos. Mas nem só os produtos de trabalho dos projetos são ICs. Os produtos de trabalho organizacionais, se críticos, também devem ser encarados como tal. Como exemplo, podemos citar as normas, padrões, guias, templates e tudo aquilo que compõe a documentação dos processos.

Mas como é feita a identificação desses ICs? Lembra do catálogo que mencionamos anteriormente? Pois é justamente lá. Se você não puder implementar um catálogo sistematizado, terá que criar algum documento escrito com essa informação. Qual informação? Nome padronizado dos artefatos, nível de controle, local de armazenamento, papéis responsáveis pelo artefato e assim por diante.



Baselines





Os resultados esperados GCO 1 e 2 falam de ICs em quaisquer níveis de controle, mas do GCO 3 em diante vamos tratar de ICs em um nível de controle muito especial: colocados sob baseline (para mais informações, consulte o quadro “O que é baseline?”).

PENSE...


QUANTO TEMPO
VOCÊ GASTARIA
PARA DESENVOLVER
COBRANÇA COM BOLETOS
BANCÁRIOS PARA
APENAS UM BANCO
NO SEU SOFTWARE

COBREBEMX

-  56 BANCOS E MAIS DE 430 CARTEIRAS DE COBRANÇA PARA IMPRESSÃO E/OU ENVIO DE BOLETO BANCÁRIO POR EMAIL;
-  GERAÇÃO DE BOLETOS ON LINE;
-  GERAÇÃO E LEITURA DE ARQUIVOS (REMESSA/RETORNO) NOS PADRÕES FEBRABAN E CNAB;
-  MAIS DE 40 EXEMPLOS EM DIVERSAS LINGUAGENS DE PROGRAMAÇÃO

DOWNLOADS E INFORMAÇÕES EM WWW.COBREBEM.COM



Tecnologia

O que deve ser colocado sob baseline? O MPS.BR não responde. Entretanto, como se trata do nível de controle mais rigoroso, é de se imaginar que sejam colocados sob baseline apenas ICs muito, mas muito críticos. Pense nos seus artefatos mais importantes e no principal produto do seu trabalho: o código fonte. Esses são fortes candidatos a comporem a sua baseline. Especificação de requisitos, diagramas de projeto detalhado e acordos de nível de serviço são outros fortes candidatos.

As baselines estão diretamente relacionadas com o seu sistema de controle de versões. Baselines de projetos geralmente são implementadas na forma de *branches*. Enquanto que baselines de sistemas geralmente são implementadas via mecanismos de *tag* (rotulação) comumente oferecidos por ferramentas de controle de versão, como o CVS e o VSS.

O GCO 4 é uma consequência do GCO 3, pois trata da descrição de cada versão que é disponibilizada. Para tal, as ferramentas de controle de versão oferecem campos para descrever textualmente um dado IC ou um pacote de ICs quando da sua disponibilização (check-out, commit, etc.).

Integre as funções do sistema

Até aqui percebemos que o controle de versões é o cerne da GCO. Mas não é tudo. O GCO 5 trata justamente da integração do controle de versões com o controle de modificações. Apesar de não ser a principal função do sistema de GCO, talvez seja a mais difícil de implementar. Muitas empresas já realizam muito bem o controle de versões, mas poucas conseguem integrar os seus sistemas de controle de modificações com o de versões. Isso significa dizer que algumas empresas não sabem dizer, de pronto, quais versões surgiram de um pedido de modificação, ou vice-versa. Estamos aqui falando de rastreabilidade bidirecional – algo que vai lhe ser muito útil no processo GRE.

Só que o GCO 5 vai além de simplesmente documentar as modificações e prover a rastreabilidade. Esse resultado esperado tem a ver também com o controle das modificações. Quando falamos em controle, estamos querendo dizer que existe um processo formal de aprovação de modificações que envolve desde a documentação da modificação, passando pela análise de impacto, avaliação das modificações e verificação para saber se o que foi implementado está de acordo com o pedido. Nesse processo, uma das coisas mais importantes é a definição dos papéis que serão responsáveis por cada passo. A ferramenta vai apenas facilitar o trabalho das pessoas que assumirem esses papéis.

Tentaremos descrever em síntese como poderia ocorrer esse processo de controle de modificações. Começa com alguém documentando uma modificação qualquer e sua necessidade. Em seguida, alguém vai verificar quais funcionalidades são afetadas por essa modificação e avaliar o impacto. A avaliação de impacto vai servir de subsídio para alguém aprovar ou não a modificação. Uma vez aprovada, considerando que seja uma pequena alteração, o analista vai fazer check-out dos ICs afetados pela alteração. Nesse

momento, o check-out será vinculado à modificação. Após as alterações, o analista vai fazer um check-in dos ICs. Depois disso, as alterações serão revisadas por alguém e, uma vez testadas e aprovadas, a liberação é autorizada. Feita a liberação, as novas versões dos ICs passam a integrar a baseline do projeto ou sistema.

Bom, estamos quase concluindo a análise dos resultados esperados. Ainda faltam os GCOs 6 e 7.

Não vamos esmiuçar aqui o GCO 6, pois ele é uma consequência do GCO 5. O GCO 6 trata da segurança da execução do GCO 5, o que é inerente ao processo de controle de modificações – ainda mais quando estamos tratando do nível de controle mais formal que é colocar sob baseline.

Auditorias de configuração

Mas o que dizer do GCO 7, que fala de auditoria da configuração? Este é um assunto ainda polêmico. Alguns avaliadores consideram essa auditoria como parte do processo Garantia de Qualidade. O próprio Guia de Implementação dá margem para isso, quando menciona, no último parágrafo de GCO 7, que as verificações podem ser feitas em parceria com a equipe de qualidade. Se assim for, basta implementarmos o processo Garantia da Qualidade. Já outros avaliadores encaram a auditoria como algo adicional à garantia da qualidade. Seja como for, a sua empresa vai ter que providenciar pessoas para executar as verificações exigidas por esse resultado.

É exigido pelo MPS.BR que essas auditorias sejam realizadas objetivamente. O que é isso? O guia podia ter sido mais claro, mas, nesse caso, objetivamente é sinônimo de independentemente. Ou seja, as pessoas que participaram do desenvolvimento do projeto não podem ser as mesmas que farão as auditorias. Objetivamente também significa que as auditorias devem ser o menos subjetivas possível. Isso envolve utilizar métodos objetivos, tais como questionários fechados e check-lists. O ideal é que as auditorias sejam feitas de forma automatizada. Contudo, isso é muito difícil de implementar para as auditorias chamadas funcionais, que têm o objetivo de verificar a qualidade do conteúdo dos ICs.

Na prática você terá que fazer o seguinte. No plano de GCO, defina as atividades de auditoria, os papéis responsáveis pela sua execução e o momento em que as auditorias serão realizadas. O foco das auditorias é, sem dúvida, as baselines de projeto e as de sistema. Daí, podemos inferir que o melhor momento para a execução das auditorias é logo após a atualização dessas baselines e antes das liberações. Se um projeto for desenvolvido em cascata, isso significa dizer que ele terá apenas uma baseline e uma liberação, ou seja, apenas uma auditoria de baseline. Já projetos desenvolvidos de forma iterativa ou incremental, podem gerar várias baselines e, portanto, talvez tenham várias auditorias. A documentação do processo GPR pode definir que essas atividades de auditoria sejam incluídas no cronograma dos projetos.

O MPS.BR também exige que as não conformidades encontradas nas auditorias sejam acompanhadas até a sua

conclusão. É por isso que é tão importante ter uma equipe de auditoria independente dos projetos. Do contrário, as atividades de auditoria vão onerar ainda mais os projetos por roubar-lhes mão de obra. As auditorias podem ser onerosas, mas o que se quer é garantir a qualidade das baselines que, como o próprio termo já indica, são a base do desenvolvimento.

Podem ser feitas auditorias em ICs que não fazem parte de baseline? Claro! Isso é até desejável. Entretanto, isso pode se tornar muito oneroso e a auditoria de baselines já basta para o grau de maturidade exigido no Nível F. Ao passo que sua empresa for sistematizando as atividades de auditoria, pode-se aumentar o seu escopo de verificação.

Próximos passos

Como o processo GCO não está sozinho no Nível F, os próximos passos, sem dúvida, estarão relacionados com os processos de Garantia da Qualidade (GQA) e Medição (MED).

O cruzamento entre GCO e MED é implementado na forma do RAP 4. Já o cruzamento entre GCO e GQA está na forma dos RAPs 10 e 14.

Infelizmente, não se pode considerar aqui esses cruzamentos, pois seria necessário destrinchar vários conceitos de qualidade e medição. Mas sem dúvida eles devem ser seus próximos desafios. Na prática, você vai se reunir com as pessoas responsáveis pela implementação desses processos e acordar as atividades de medição e qualidade que devem permear o processo GCO, e vice-versa.

Falando só um pouco sobre medição, você vai precisar de indicadores para esse processo. Uma dica é utilizar os resultados das auditorias de GCO para calcular indicadores exigidos pelo processo Medição.

Conclusão

Vimos aqui alguns aspectos práticos para começar uma implementação do processo GCO a partir do zero. É importante que todos os demais processos já comecem se preocupando com GCO, mesmo no Nível G, para se evitar dificuldades futuras. Vimos também a importância de construir um bom alicerce para o processo por meio dos RAPs. Com esses pontos em mente, fica muito mais fácil implementar esse processo tão importante que se cruza com todos os demais processos. Em artigos futuros, vamos ver quais são as evidências típicas dos resultados esperados e RAPs de GCO. ●

Links

Softex > MPS.BR > Guias

http://www.softex.br/mpsbr/_guias/default.asp

IBM Rational ClearCase

<http://www-01.ibm.com/software/awdtools/clearcase/>

IBM Rational ClearQuest

<http://www-01.ibm.com/software/awdtools/clearquest/>

Synergy ALM

http://www.eclipseplugincentral.com/Web_Links-index-req-viewlink-cid-1173.html

Aldon Application Lifetime Management Suite

<http://www.aldon.com/prod/alm/ov/>

TD/OMS

<http://www.remainsoftware.com/>

Referências

PRESSMAN, R. S. Software Engineering: a practitioner's approach. Mc Graw Hill Higher Educational, 6ª. Edição. 2005.

MAHLER, A. Variants: Keeping things together and telling them apart. In Configuration Management, Vol. 2 of Trends in Software, Wiley, New York, 1994.

BERSOFF, E. H.; Henderson, V. D. e Siegel, S.G. Software Configuration Management: A tutorial. Los Alamos, Califórnia. IEEE Computer. v.12, n.1, 1979.

IEEE for Software Configuration Management Plans. 1998.

TUSCANY, P.A. Software development environment for large switching projects. In Proceedings of Software Engineering for Telecommunications Switching Systems Conference, 1987.

PACHECO, R. F. Uma Forma de Implantação de Gerenciamento de Configuração de Software em Empresas de Pequeno Porte. Dissertação (Mestrado) – Instituto de Ciências Matemáticas de Computação, Universidade de São Paulo, São Carlos, 1997.

SOFTEX, 2006b, MPS.BR – Melhoria de Processo do Software Brasileiro – Guia de Implementação (Versão 1.1), Associação para Promoção da Excelência do Software Brasileiro.

ISO, 1995b, ISO/IEC 12207 – Information technology – Software life cycle processes, International Organization for Standardization.

Dê seu feedback sobre esta edição!

A Engenharia de Software Magazine tem que ser feita ao seu gosto.

Para isso, precisamos saber o que você, leitor, acha da revista!

Dê seu voto sobre este artigo, através do link:

www.devmedia.com.br/esmag/feedback

