

ACH2006 - Engenharia de Sistemas de Informação I

Gerência de Configuração de Software
(GCS)

Problema

- Requisitos mudam e novos requisitos devem ser incorporados.
- Muitas versões podem estar em desenvolvimento ao mesmo tempo.
- Correções são feitas e novas versões incorporam correção de defeitos.
- Adaptações de hardware e sistemas operacionais são necessários.

Problema

- Como tratar estes problemas?
 - Fazendo gerência de configuração de software (GCS).

Objetivo da GCS

- armazenar diferentes versões de arquivos;
- como construir *releases* do sistema a partir de determinadas versões de arquivos;
- rastrear os *releases* do sistema;
- controlar mudanças: aprovar/rejeitar, analisar impacto e custo, implementar e validar.

Problemas sem GCS

- Perda de rastreabilidade:
 - Quais mudanças foram incorporadas na release X.Y?
 - quais são as versões dos arquivos necessários para gerar a release X.Y?
- Esforços repetidos: mudanças podem ter de ser repetidas porque simplesmente não se sabe aonde estão as versões corretas.
 - Como recuperar uma versão em que o requisito RF 2.3 funcionava?

Gerência de Configuração

- Os procedimentos de gerenciamento de configuração definem como registrar e processar mudanças de sistemas, como relacioná-las aos componentes do sistema e os métodos usados para identificar diferentes versões dele.

Observação importante: GCS é impossível de realizar sem o apoio de ferramentas.

Definições

- O Gerenciamento de Configuração de Software (GCS) visa identificar a configuração de um sistema em diferentes pontos no tempo, com o propósito de sistematicamente controlar alterações e manter a integridade e a rastreabilidade da configuração ao longo do ciclo de vida de software [IEEE 2004].
- O GCS é o desenvolvimento e o uso de padrões e procedimentos para o gerenciamento de sistemas de software em desenvolvimento [Sommerville, 2007].

Observações

- GCS não é um aspecto técnico de construção do sistema, mas um aspecto gerencial.
- É considerado em alguns casos parte da Gerência de Qualidade do processo de desenvolvimento.
- Possui um aspecto técnico que é a configuração e apoio ao uso de ferramentas.

Observações

- A implementação de GCS é essencial para a certificação de qualidade para os padrões ISO 9000, CMM e CMMI.
- O IEEE possui um padrão para GCS como possui para elaboração do documento de requisitos.

Idéias Básicas

- O software é inicialmente liberado para equipe de desenvolvimento para a garantia de qualidade do sistemas. A equipe de GQ (QA – Quality Assurance) verifica se o sistema possui qualidade aceitável.
- Uma vez aprovado, ele se transforma em um sistema controlado, ou seja, as mudanças que sofrer devem ser **acordadas e registradas** antes de serem implementadas.

Idéias Básicas

- Está é a idéia original de GCS que não leva em consideração o carácter interativo/incremental das maioria dos processos de desenvolvimento de software.
- Há intrinsecamente associado a esta idéia básica um processo de desenvolvimento em cascata subjacente.
- A GCS precisa, no entanto, ser analisada, e posta em prática, considerando o fato que a maioria do desenvolvimento atual é feito de maneira incremental/interativa.

Mais definições

- **Configuração básica ou *baseline***: um conjunto de itens de configuração que constituem os artefatos de software (documentos, diagramas, código, massa de teste, scripts de teste, etc.) que são controlados pela GCS.
- **Item de configuração**: um artefato de software que faz parte de uma configuração básica controlado pela GCS.
- Artefatos de software como documentos de requisitos, diagramas, código-fontes, *scripts* de teste, documentos de projetos etc. são candidatos a serem itens de configuração.

Build

- É uma construção do sistema pronta para executar.
- O *build* implica que todos os arquivos foram compilados e o executável gerado.
- Um *build* é construído com várias finalidades. Uma delas é para executar a massa de testes de testes e verificar/validar o sistema.
- Outra é para ser incluído em uma release do sistema para ser entregue ao cliente/usuário.

Release

- É uma dada composição executável do software, incluindo os itens de configuração da configuração básica do software naquele momento.
- Normalmente, uma *release* é entregue para o cliente/usuário para validação e contém um conjunto de requisitos do sistema implementados.

Release

- Em uma abordagem incremental, as *releases* representam diferentes composições do software com um número crescente de requisitos implementados até, finalmente, todos os requisitos estarem implementados e validados pelo usuário.

Versão

- Termo com várias acepções:
 - uma versão de arquivo ou
 - uma release do software.

GCS e o processo de desenvolvimento

- Em cascata:
 - Software entregue para equipe de GCS;
 - GCS assume responsabilidade pela construção do sistema e gerenciamento dos testes;
 - Defeitos descobertos durante os testes são encaminhados para a equipe de desenvolvimento;
 - Depois das correções, a equipe de desenvolvimento entrega uma nova versão do componente reparado para a equipe de garantia de qualidade.
 - Se qualidade aceitável, ele pode se tornar uma nova configuração básica (*baseline*) para o desenvolvimento futuros do sistema.

GCS e o processo de desenvolvimento

- Para desenvolvimento incremental:
 - A organização de desenvolvimento estabelece determinada hora para entrega dos componentes do sistema. Os componentes podem estar incompletos mas devem prover algumas funcionalidades básicas que podem ser testadas.
 - Uma nova versão do sistema é construída baseada nesses componentes, pela compilação e ligação deles para formar um sistema completo.

GCS e o processo de desenvolvimento

- Para desenvolvimento incremental:
 - O sistema é depois entregue para a equipe de testes, que cuida de um conjunto predefinido de testes de sistema. Ao mesmo tempo, os desenvolvedores estão ainda trabalhando nos seus componentes, incluindo funcionalidades e reparando defeitos descobertos nos testes anteriores.
 - Defeitos descobertos durante os testes de sistema são documentados e devolvidos aos desenvolvedores do sistema. Eles reparam esses defeitos na versão seguinte do componente.

Vantagens

- Chances de encontrar problemas oriundos de interações de componentes no processo aumentam.
- Construções diárias encorajam testes minuciosos de componentes;
- Desenvolvedores são pressionados psicologicamente a não tornar disponível módulos pouco testados;

Desvantagens

- Requer gerenciamento de mudanças para encontrar problemas em cada build diário;
- Gera um grande número de builds;

Planejamento de gerência de configuração

- Sommerville:
 - Define o que será gerenciado (itens de configuração) e o esquema que se deve usar para identificar essas entidades.
 - Estabelece quem é o responsável pelos procedimentos de gerência de configuração e pela submissão de itens controlados por a equipe de gerenciamento de configuração.
 - Define as políticas de GCS que todos os membros da equipe devem adotar para o controle de mudanças e gerenciamento de versões.

Planejamento de gerência de configuração

- Sommerville:
 - Especifica as ferramentas que se deve usar para GCS e o processo para uso dessas ferramentas.
 - Descreve a estrutura do banco de dados de configuração usada para registrar as informações de configuração e as informações que devem ser mantidas nesse banco de dados (registros de configuração).
 - Dependendo do tamanho do projeto, a criação deste banco de dados ou a compra de uma ferramenta que realiza estas tarefas não se justifique. Porém, é necessário que as informações de configuração sejam armazenadas para recuperação posterior.

Observações

- Importante é definir responsabilidades:
 - quem faz artefato de software;
 - quem revisa; quem entrega;
 - quem faz não precisa ser quem entrega.
- O gerente de projeto ou os líderes da equipe são freqüentemente responsáveis por todos os documentos produzidos pela equipe.
- Faz parte do planejamento definir quais itens de configuração serão armazenados e rastreados, isto é, controlados.

Itens de Configuração

- Precisam ser identificados:
 - deve-se atribuir um nome geral para todos os artefatos sob controle de configuração, por exemplo, o nome do projeto.
 - Deve-se atribuir “subnomes” divididos por módulos do projeto e assim por diante até o documento propriamente dito.
 - O esquema de identificação é parte do planejamento da GCS.
 - Mais comum: esquema hierárquico baseado em diretórios, e.g., esquema Java de diretórios.

Problemas

- Este esquema dificulta o reuso de artefatos de software, uma vez que eles são extremamente vinculados a um projeto.

Bancos de dados de configuração

- Registra todas as informações relevantes sobre as configurações de sistema e os itens de configuração.
- Útil para:
 - avaliação de impacto de mudanças;
 - relatórios sobre o processo de GCS;
 - conter informações sobre usuários de componentes, clientes de sistemas, plataformas de execução, mudanças propostas etc.

Bancos de dados de configuração

- Pode ser utilizado para responder as seguintes questões:
 - Quais clientes pegaram um produto de uma versão específica do sistema?
 - Qual configuração de hardware e de sistema operacional é necessária para rodar determinada versão do sistema?
 - Quantas versões de um sistema foram criadas e quais eram as datas da sua criação?
 - Quais versões de um sistema podem ser afetadas se um componente específico for alterado?
 - Quantas solicitações de mudanças estão pendentes para uma versão específica?
 - Quantos defeitos reportados existem em uma versão específica?

Bancos de dados de configuração

- Ferramentas como:
 - CVS e Subversion; Bugzilla e SourceForge apóiam algumas das tarefas acima. São públicas (“de grátis”), mas não são integradas. Suporte a estas ferramentas não é trivial.
 - Ferramentas CASE integradas para GCS existem e são caras: ClearCase e ClearQuest – IBM Rational. O suporte pode também ser pago.

Gerência de Mudanças

- Mudanças são fatos da vida...
- Procedimentos de GCS são necessários para que as mudanças sejam realizadas ou “desrealizadas”, se necessário.
- Procedimentos de GCS apoiado por ferramentas são necessários.

Gerência de Mudanças

- Implica:
 - análise de custo e benefício;
 - aprovação/rejeição das mudanças propostas;
 - rastreabilidades dos artefatos que serão mudados ou influenciados pelas mudanças.

Gerência de Mudanças

- Ocorre:
 - Depois que existe uma configuração básica – *baseline*.
 - Métodos ágeis: requisição de mudança é informal – o cliente/usuário é residente e pode aprovar ou não a mudança.
 - Tradicional: é preciso preencher um documento de requisição de mudança (*Change Request Form*);

Gerência de Mudanças

- Inclui recomendações, custos estimados e datas de quando foi solicitada, por quem, data de aprovação, implementada e validada.
- Pode incluir uma seção em que uma descrição da possível implementação da mudança é descrita.
- Se aprovada a mudança é realizada;
- Um comitê de controle de mudanças (*Change Control Board*) deve revisar e aprovar todas as solicitações de mudanças.
- CCB simplificado: somente o gerente de projeto.

Gerência de Mudanças

- Software *genéricos* (e.g., MS Office):
 - usuários relatam defeitos ou sugestões de mudanccas por *email* ou *websites*.
 - Modificações são analisadas e decididas dependendo da prioridade, conveniência de *marketing* e custos.

Registro de Mudanças

- Normalmente, itens de configuração contêm um relato das mudanças realizadas e dos motivos que levaram às modificações.
- Por exemplo, em arquivos fonte é comum colocar um cabeçalho na forma de comentário que inclui as modificações realizadas. O mesmo pode ser feito com páginas Web.

Registro de Mudanças

- Em documentos publicados, é comum incluir uma página preliminar que descreve as mudanças que o documento sofreu.
- É possível escrever um *script* que analisa as datas das versões dos itens de configuração e produz um relatório de mudanças.

Exercícios

- Defina os termos configuração básica e item de configuração.
- Considerando o sistema que o seu grupo está desenvolvendo faça o planejamento da Gerência de Configuração necessária para a condução do projeto. Defina os itens de configuração que serão controlados e quando serão controlados, projete um ambiente de desenvolvimento para gerência de configuração indicando os recursos necessários (recursos de *hardware* e software necessários; você pode, por exemplo, indicar o tipo de ferramentas necessárias).
- Compare os termos versão, build e release.

Bibliografia

- SOMMERVILLE, I. Engenharia de Software, Addison-Wesley Brasil, 6ª Edição – 2003, 608 páginas. (Versão mais recente em inglês na biblioteca).