Pontifícia Universidade Católica do Rio de Janeiro

Quinto Resumo de Projeto de Engenharia de Software

Grupo Open Source:

Eduardo Telles
Giovani Tadei
Marco Aurélio
Renata Monique

Primeira Aula → Gerência (Princípios e Ferramentas) (09/09/2004)

1) O que Aprendi

A idéia principal desta aula, foi expor alguns modelos de controle de qualidade (continuação das aulas passadas), mostrar as atividades de gerência e as ferramentas que auxiliam a mesma.

Sobre os modelos de controle de qualidade foi citado o PDCA(plan, do, check e act). O método PDCA se baseia no controle de processos para melhoria contínua. Foi desenvolvido na década de 30 pelo americano *Shewhart*, mas foi *Deming* seu maior divulgador, ficando mundialmente conhecido ao aplicar tais conceitos de qualidade no Japão (MILET, 1993; BARRETO, 1999). O PDCA pode ser explicado como sendo uma seqüência lógica de 4 etapa que se repetem.

1 etapa - P (Plan = Planejar)

Definir o que queremos, planejar o que será feito e possíveis mudanças, estabelecer metas, e definir os métodos que permitirão atingir as metas propostas. No caso de desenvolvimento de um sistema de informação, esta atividade pode corresponder ao planejamento do sistema.

2 etapa - D (Do = Executar)

Tomar iniciativa, educar, treinar, implementar, executar o planejado conforme as metas e métodos definidos. No caso de desenvolvimento de um sistema de informação, esta atividade pode corresponder ao desenvolvimento e uso do sistema.

3 etapa - C (Check = Verificar)

Verificar os resultados que se está obtendo, verificar continuamente os trabalhos para ver se estão sendo executados conforme planejados. No caso de desenvolvimento de um sistema de informação, esta atividade pode corresponder aos testes, análise das informações geradas e avaliação de qualidade do sistema.

4 etapa - A (Act = Agir)

Fazer correções de rotas se for necessário, tomar ações corretivas ou de melhoria, caso tenha sido constatada na fase anterior a necessidade de corrigir ou melhorar processos. No caso de desenvolvimento de um sistema de informação, esta atividade pode corresponder aos ajustes, implementações e continuidade do sistema.

Na 3^0 etapa (check) ocorre uma verificação dos erros ocorridos na 2^0 etapa (do) previstos na 1^0 etapa (plan) para que os mesmo não voltem a acontecer. Na transição da 3^0 para a 4^0 etapa podemos identificar o feedback do processo. Tal feedback pode ser caracterizado como contínuo, pois gera "aprendizado", ou seja, na etapa de check são armazenados todas as informações sobre as ações executadas e com isso geramos uma memória que poderá ser útil futuramente.

A identificação das etapas em modelo PDCA em cascata fica da seguinte maneira: A 1^0 etapa englobam o planejamento inicial e a definição dos requisitos, a 2^0 etapa pode

ser caracterizada pelo design lógico, design físico e a codificação, a 3^0 etapa seriam os teste e a manutenção a 4^0 etapa.

Na segunda parte da aula foi introduzido a idéia básica de gerência e suas atividades principais. Sobre gerência podemos citar o Project Management Body of Knowledge (PMBOK) que seria a consolidação das melhores práticas e métodos usados para gerenciar um projeto. Trata-se de um produto do Project Management Institute (PMI), um dos mais conceituados institutos mundiais sobre gerência.

Segundo o PMI, um gerente de projeto deverá estar atento a todo o contexto que dirá respeito à sua gerência, ao ciclo de vida (divisão por fases), aos stakeholders (os envolvidos direta e indiretamente com o projeto), às influências organizacionais e às influências sócio-econômicas. Destacam-se como habilidades gerenciais a liderança, a comunicação, a negociação, a resolução de problemas e a influência na organização.

Basicamente a gerência consiste em 4 atividades principais que são organizar, planejar,coordenar e controlar.

Na parte de organização o gerente atribui papéis e funções a cada um dos membros da equipe de desenvolvimento, para o tal é utilizado normalmente um organograma(o organograma é uma representação gráfica que apresenta, ao mesmo tempo, a divisão do trabalho e as relações de poder no âmbito de uma organização). Além disso o gerente deve tomar decisões relativas a infra-estrutura do projeto (espaço físico e plataformas de software e hardware) .

Na etapa de planejamento o gerente deve traçar objetivos, descrever o objetivo final através de sub-objetivos e marcar milestones. Milestones seria um acompanhamento que deve ser feito periodicamente por parte do gerente durante um projeto, para que o mesmo fique sabendo continuamente como anda o projeto. Ou seja, em uma equipe onde a produção não é satisfatória seria interessante utilizar milestones mais freqüentes. Cabe salientar que o planejamento é individual para cada projeto. Mais adiante serão expostas as ferramentas de planejamento (CVS, pert/cpm e xplanner).

A coordenação consiste em motivar, ajudar e liderar a equipe para que a mesma siga as metas traçadas anteriormente com os recursos disponíveis.

Para garantir que a equipe está seguindo o plano traçado anteriormente, o gerente deve controlar a diferença entre o plano traçado e a realidade. Invariavelmente um projeto sofre desvios. Tais desvios podem ocorrer por diversas razões como imprevistos e problemas na coordenação.

Também é tarefa do gerente gerenciar as configurações do sistema de software (documentação, ferramentas, bibliotecas ...). Com a importância da gerencia de configuração num projeto, houve a necessidade de criação de ferramentas de auxilio e a criação do cargo de gerente de configuração.

No processo de produção baseado em extreme programming o gerente tem o seu papel dividido em duas partes: Tracker e o Coach.

Tracker:

- Coletar sinais vitais do projeto (métricas) uma ou 2 vezes por semana
- Manter todos informados do que esta acontecendo
- Tomar atitudes sempre que as coisas parecerem ir mal
- Checar prazos

- Sugerir sessões de CRC sempre que necessário

CRC é uma "linguagem" para modelagem de classes como o diagrama de classes do UML, entretanto, ele é bem menos rígido, é feito com cartões, e o objetivo é mais "explorar" várias soluções do que documentá-las.

Coach:

- Ajudar no que for necessário
- mentor
- tem conhecimento técnico
- Garantir que o projeto permaneça extremo (Significa continuar dentro das práticas do XP).
- realizar pair programming

A idéia básica do pair programming seria que todo o código de produção deve ser desenvolvido por duas pessoas trabalhando juntas no mesmo computador.
Para funcionar, os dois programadores devem entrar em sintonia: enquanto um digita com o teclado o outro pensa em funcionalidades, fica atento a bugs e cuida para que a codificação esteja no rumo certo.

Após a II Guerra Mundial, a complexidade dos projetos e a retração do trabalho e de suprimentos demandou novas estruturas organizacionais. Diagramas de rede complexos (PERT – Program Evaluation and Review Technique), e métodos de análise do caminho crítico (CPM – Critical Path Method) foram introduzidos, oferecendo aos gerentes maior controle sobre projetos. Rapidamente essas técnicas foram difundidas entre gerentes que procuravam novas estratégias e ferramentas de gerenciamento que permitissem o desenvolvimento de projetos em um mundo competitivo e de mudanças rápidas. Sobre as ferramentas de gerencia de projetos podemos citar pert/cpm, xplanner e o CVS.

O Pert (Project Evaluation Review and Technique) é instrumento gráfico de gestão que facilita a planificação, programação e controlo de projectos complexos, pondo em evidência a relação existente entre as diferentes actividades. Podemos caracaterizar o Pert como sendo um tipo específico de diagrama de rede para projetos que é raramente utilizado hoje em dia. Um diagrama de rede de projeto é um esquema de apresentação das atividades do projeto e dos relacionamentos lógicos (dependências) entre elas.

Em termos de calculos de duração de projetos existem diferenças superficiais entre o pert e o cpm. O pert difere fundamentalmente do CPM por usar distribuição de médias (valor esperado) em vez do valor mais provável, originalmente usado no CPM. O pert propriamente dito é muito pouco utilizado atualmente, embora as estimativas similares do PERT (PERT-like) sejam freqüentemente usadas nos cálculos de CPM.

Com as pequenas diferenças entre o pert eo o cpm essas dusa ferramentas se juntaram. Algumas caracteristicas fundamentais do pert/cpm são: nós representam eventos no projeto, vértices representam tarefas com duração associada, tarefas paralelas não dependem de recursos comuns, tarefas dependentes utilizam recursos em comum (não podem ser executadas ao mesmo tempo). A deependencia entre eventos determina o caminho crítico, ou seja, o comprimento do aminho crítico d dado pela soma das durações das tarefas.

Xplanner é uma ferramneta de planejamneto e acompanhamento para adeptos do extreme programming. Apartir da escolha de um periodo de iteração, onde um conjunto de requisitos será implementado, o cliente válida ou não os requisitos implementados neste período. As principais caracteristicas do Xplanner são os users storys (com o cliente sempre disponivel para tirar eventuais dúvidas) e as tarefas (definidas pelo programador).

O CVS(Concurrent Versions System**)** é uma ferramneta de controle de versão. Com o CVS, é possível que grupos de pessoas trabalhem simultaneamente no desenvolvimento de software e outros produtos baseados em arquivos como, por exemplo, manuais, apostilas, dados de configuração, dentre outros.

Existe um repositório central com as versões mais recentes dos arquivos. Com isto, é possível criar uma cópias desses arquivos e, caso no futuro esses arquivos sejam atualizados, o CVS se encarregará de atualizar as cópia quando for necessário fazê-lo. Se novas versões destes arquivos forem colocadas no CVS nesse período de modificação, a atualização tentará o máximo possível combinar as alterações do repositório com as feitas localmente.

Caso haja coincidência nas alterações e algum conflito seja criado pelas mudanças concorrentes, o CVS notificará o conflito e será necessário alguma intervenção humana para resolver este problema.

Vantagens trazidas pelo uso do CVS:

- Ajuda a rastrear erros no software. Por exemplo, quando o código é modificado, é comum que o programador introduza erros que somente são detectados muito tempo após a alteração.
- Com CVS, é possível recuperar uma versão anterior que funcione para identificar a causa do erro.
- Facilita a colaboração entre desenvolvedores, permitindo que dois ou mais desenvolvedores trabalhem de forma independente sobre o mesmo código fonte.
- Permite identificar e resolver conflitos nos fontes.
- Facilita o gerenciamento de releases do código.
- Permite verificar a produtividade de um programador, através do histórico de modificações.
- Possibilidade de utilizar as informações do repositório do CVS como fonte de métricas que subsidiem o processo de desenvolvimento de software nas organizações

O CVS é um padrão para o processo de desenvolvimento de Software Livre, sendo utilizado por milhares de desenvolvedores ao redor do mundo no desenvolvimento de software. O site SourceForge.net talvez seja o melhor exemplo da utilização do CVS como ferramneta ao desenvolvimento concorrente e por equipes distribuídas geograficamente. Atualmente existem mais de 50.000 projetos cadastrados neste site e um volume de mais de 500.000 desenvolvedores.

Excelente resumo!

2) O que não entendi

Não houve.

3) O que duvido

Não houve.

Bibliografia:

http://www.utp.br/informacao/si/si_ciclo%20PDCA%20e%205S.htm

http://www.pr.gov.br/batebyte/edicoes/2003/bb130/abordagem.shtml

http://www.activedelphi.com.br/modules.php?op=modload&name=News&file=article&sid=118

http://www.jrothman.com/Papers/7ICSQ97.html

http://www.xispe.com.br/index.html

http://www.dextra.com.br/empresa/boletim/0310-04/04gestao.htm