

## Implementação

A finalidade da implementação é:

- definir a organização do código em termos de subsistemas de implementação organizados em camadas
- implementar classes e objetos em termos de componentes (arquivos-fonte, binários, executáveis e outros)
- testar os componentes desenvolvidos como unidades
- integrar os resultados produzidos por implementadores individuais (ou equipes) ao sistema executável

A disciplina Implementação limita o seu escopo a como classes individuais devem ser testadas em unidade. O teste do sistema e o teste de integração são descritos na disciplina Teste.

## Teste

A disciplina Teste atua em vários aspectos como uma provedora de serviços para as outras disciplinas. O teste enfatiza principalmente a avaliação da qualidade do produto, realizada através de várias práticas centrais:

- Localizar e documentar defeitos na qualidade do software.
- Avisar de forma geral sobre a qualidade observada no software.
- Validar as suposições feitas nas especificações de design e requisito através de demonstração concreta.
- Validar as funções do software conforme projetadas.
- Verificar se os requisitos foram implementados de forma adequada.

Uma diferença interessante entre o Teste e as outras disciplinas do RUP é que a principal finalidade do Teste é localizar e expor os pontos fracos do software. O interessante é que, para obter o máximo benefício, ele precisa de uma filosofia geral diferente da usada nas disciplinas Requisitos, Análise e Design, e Implementação. A diferença sutil é que enquanto as outras disciplinas enfatizam a abrangência, o Teste enfatiza a deficiência. Um bom esforço de teste é direcionado por perguntas do tipo "Como esse software pode ser dividido?" e "Em que situações este software pode deixar

de funcionar da maneira prevista?". O teste desafia as suposições, os riscos e as incertezas inerentes ao trabalho de outras disciplinas, tratando essas questões por meio de uma demonstração concreta e uma avaliação imparcial. O desafio é evitar dois extremos potenciais: uma abordagem que não avalie o software de forma adequada e efetiva, expondo seus problemas e pontos fracos, e uma abordagem que seja inapropriadamente negativa ou destrutiva. Com a adoção de uma abordagem tão negativa, talvez você nunca considere aceitável a qualidade do software, e provavelmente alienará o esforço de Teste de outras disciplinas.

Com base nas informações apresentadas em várias pesquisas e ensaios, o teste de software é responsável por 30% a 50% dos custos totais de desenvolvimento do software. Portanto, talvez seja surpreendente notar que a maioria das pessoas acredita que o software não é bem testado antes de ser fornecido. Essa contradição baseia-se em algumas questões essenciais.

Primeiro, o teste de software é bastante difícil. Não é possível quantificar as diversas maneiras como um programa pode se comportar. Segundo, o teste é normalmente realizado sem uma metodologia clara e, portanto, os resultados variam de acordo com o projeto e a organização. O sucesso é um fator ligado principalmente à qualidade e às habilidades dos indivíduos. Terceiro, as ferramentas de produtividade são usadas de forma insuficiente, tornando gerenciáveis os aspectos trabalhosos do teste. Além da falta de execução de testes automatizados, vários esforços de teste são conduzidos sem ferramentas que permitam o gerenciamento eficaz de Dados de Teste e Resultados de Teste abrangentes. Enquanto a flexibilidade de uso e a complexidade do software tornam o teste completo uma meta impossível, uma metodologia bem-concebida e o uso de ferramentas avançadas podem ajudar a melhorar a produtividade e eficácia do teste de software.

Em sistemas de "segurança crítica", nos quais uma falha pode causar danos a pessoas (como, por exemplo, controle do tráfego aéreo, direção de mísseis ou sistemas de assistência médica), um software de alta qualidade é essencial para o sucesso desses sistemas. Em um sistema MIS típico, seu grau de risco pode não ser imediatamente óbvio, mas é provável que o impacto de um defeito possa acarretar ao negócio que utiliza o software uma despesa considerável, devido a perda de receita ou possíveis custos legais. Nesta "era da informação", com o aumento da demanda por serviços

eletrônicos pela Internet, vários sistemas MIS são agora considerados como de "missão crítica"; ou seja, as empresas não podem realizar as suas funções e sofrem grandes prejuízos com a ocorrência de falhas.

Uma abordagem contínua da qualidade, realizada no início do ciclo de vida do software, pode diminuir significativamente o custo de conclusão e manutenção do software. Isso reduz bastante o risco associado à implantação de software de má qualidade.

## **Implantação**

A Disciplina Implantação descreve as atividades que garantem que o produto de software será disponibilizado a seus usuários finais.

A Disciplina Implantação descreve três modos de implantação de produto:

- a instalação personalizada
- o produto em uma forma "compacta"
- acesso ao software por meio da Internet

Em cada instância, a ênfase é testar o produto no local de desenvolvimento, seguido de testes beta, antes de ele ser finalmente oferecido ao cliente.

Embora o pico das atividades de implantação seja a Fase de Transição, algumas atividades ocorrem em fases anteriores ao planejamento e à preparação para a implantação.

## **Plano de gerenciamento de configuração**

O Plano de Gerenciamento de Configuração (CM) descreve todas as atividades do Gerenciamento de Controle de Configuração e Mudança (CCM) que serão executadas durante o ciclo de vida do produto ou do projeto. Ele detalha o cronograma de atividades, as responsabilidades atribuídas e os recursos necessários, como equipes, ferramentas e computadores.

O gerente de configuração disponibiliza o ambiente e a infra-estrutura geral de Gerenciamento de Configuração (CM) para a equipe de desenvolvimento do produto. A função de CM oferece suporte à atividade de desenvolvimento de produtos para que os

desenvolvedores e integradores tenham espaços de trabalho adequados para criar e testar seus trabalhos e, dessa forma, permite que todos os artefatos fiquem disponíveis para inclusão na unidade de implantação, conforme necessário. O gerente de configuração também deve assegurar que o ambiente de CM facilite a revisão do produto e as atividades de controle de mudanças e defeitos. O gerente de configuração também é responsável por redigir o Plano CM e relatar estatísticas de andamento com base nas solicitações de mudança.

O gerente de configuração deve conhecer os princípios de gerenciamento de configuração e, preferencialmente, ter experiência ou treinamento no uso de ferramentas de Gerenciamento de Configuração. Um bom gerente de configuração está atento aos detalhes. Ele deve ser assertivo para assegurar que os desenvolvedores não ignorem as políticas e os procedimentos de gerenciamento de configuração.

A finalidade do Plano CM é definir ou fazer referência às etapas e atividades que descrevem como o Gerenciamento de Controle de Configuração e Mudança é executado no desenvolvimento de um software.

### **Guia de teste**

É um registro documentado de qualquer um destes itens: decisões de controle e execução do processo, padrões que devem servir de base ou orientações de boas práticas que geralmente devem ser seguidas pelos praticantes de um projeto específico.

O papel Designer de Teste é responsável por definir a abordagem de teste e assegurar sua correta implementação. O papel envolve identificar as técnicas, ferramentas e diretrizes apropriadas para implementar os testes necessários e dar orientação sobre os correspondentes requisitos de recursos para o esforço de teste. Às vezes, este papel também é chamado de *Arquiteto de Testes*, *Arquiteto de Automatização de Testes* ou *Especialista em Automatização de Testes*. Este papel é responsável por:

- Identificar e descrever as técnicas de teste apropriadas
- Identificar as ferramentas de suporte apropriadas
- Definir e manter a Arquitetura de Automatização de Teste
- Especificar e verificar as Configurações do Ambiente de Teste Obrigatórias
- Verificar e avaliar a Abordagem de Teste

Os papéis organizam a responsabilidade de executar atividades e desenvolver artefatos em grupos lógicos. Cada papel pode ser designado a uma ou mais pessoas, e cada pessoa pode desempenhar um ou mais papéis. Ao definir o perfil do papel Designer de Teste, você deve considerar as habilidades exigidas para o papel e as diferentes abordagens que podem ser feitas para designar o papel ao pessoal.

O Guia de Teste atende a duas finalidades relacionadas, porém, com diferenças sutis:

- Para registrar ajustes (normalmente táticos) na maneira de execução do processo em um determinado projeto.
- Capturar as práticas específicas do projeto descobertas durante a execução dinâmica do processo.