

ETAPAS DE IMPLANTAÇÃO E MANUTENÇÃO DO PROCESSO DE DESENVOLVIMENTO DE SOFTWARE

IMPLANTAÇÃO

Considerando as etapas do processo de desenvolvimento de software apresentadas, o que temos até o momento?

O produto software testado em condições de migrar para o ambiente denominado “produção”, ou seja, mundo real, onde os usuários irão utilizá-lo nas suas rotinas diárias, seja na realização de um controle administrativo, seja no contexto de um processo de tomada de decisão.

A implantação é a etapa do processo de desenvolvimento de software relacionada à transferência do sistema da comunidade de desenvolvimento para a comunidade de usuários, ou seja, o sistema entra em produção no ambiente real.

O objetivo da implantação é produzir com sucesso a entrega do software para seus usuários finais, podendo cobrir vasta gama de atividades, como descritas a seguir:

A produção de releases externos do software

Inclui o lançamento de uma nova versão do software, ou resultado de uma manutenção no mesmo, sendo exigidas decisões, por parte de desenvolvedores e usuários, sobre como realizar a distribuição do produto.

A embalagem do software

Quando o software será comercializado por meio de determinada mídia, como, por exemplo, DVD.

Distribuição do software

É o processo de entrega de software para o usuário final, podendo ocorrer de forma manual, sem auxílio de ferramentas, ou automática, com auxílio de ferramentas.

Instalação do software

É um processo que inclui a instalação de todos os arquivos necessários à execução do software projetado.

Prestação de ajuda e assistência aos usuários

Quando a complexidade do software exige um serviço de atendimento aos clientes que procuram esclarecimentos sobre dúvidas relacionadas às funcionalidades ou sobre problemas técnicos relacionados ao software, e.g., serviço de *help desk*.

GESTÃO DE CONFIGURAÇÕES *VERSUS* IMPLANTAÇÃO

Vamos imaginar que estamos desenvolvendo um software utilizando o fluxo de processo iterativo e incremental, ou evolucionário, sendo esse fluxo comumente adotado atualmente. Após os testes, geramos a primeira versão e a disponibilizamos no ambiente de produção. Neste ponto, realizamos a implantação dessa versão.

O que faz a equipe desenvolvedora? Inicia o desenvolvimento da segunda versão, ou seja, mais um conjunto de requisitos é analisado em um novo ciclo iterativo. Em algum momento, soa um alerta dos usuários informando que “temos um defeito” no software em produção.

O que você faria como engenheiro de software?

Cenário 1

Terminar a versão 2 com o defeito corrigido e liberá-la para produção.

Cenário 2

Realizar a manutenção, eliminando o defeito na versão 1, liberá-la para produção e realizar a devida alteração na versão 2 em desenvolvimento.

Tecnicamente, o cenário 2 é a melhor solução, pois o usuário teria de aguardar a liberação da versão 2, convivendo por um período com o impacto negativo do defeito do software no ambiente de produção.

Neste pequeno estudo de caso, podemos destacar dois problemas para o engenheiro de software: o controle de alterações e o controle de versões. Nesta etapa, temos de considerar alguns aspectos que relacionam a gestão de configurações com a implantação de um sistema.

Gestão de configuração, de forma simplificada, é um conjunto de tarefas que visam gerenciar as alterações durante o desenvolvimento do software, sendo essa gestão aplicada em todas as etapas do processo de desenvolvimento de software. Uma das tarefas do processo de gestão de configuração é denominada “controle de alterações”, permitindo solucionar o primeiro problema apresentado anteriormente.

A Figura 15 ilustra um processo de controle de alterações, que, a partir de uma solicitação de alteração, requer a avaliação do mérito técnico, dos efeitos colaterais em potencial, do impacto global em termos de configuração e da funcionalidade e do custo da alteração. Na referida figura, o acrônimo ECO (*Engineering Change Order*) representa um pedido de alteração de engenharia.

Qual a importância deste processo para a etapa de implantação?

O processo tem de estar bem definido e ajustado à complexidade do software quando implantado, pois os defeitos serão identificados em produção, e os procedimentos de alteração estarão definidos. Lembre-se de que, em projetos complexos, a falta de um controle de alterações pode gerar o caos. Afinal, sabe-se que a fase de testes não é capaz de zerar a existência de erros.

Vamos, agora, ao segundo problema do nosso pequeno exemplo apresentado, ou seja, “controle de versões”. Neste caso, ainda no contexto do gerenciamento de configuração, podemos destacar o gerenciamento de releases.

Um release de sistema é uma versão do mesmo distribuída aos clientes. A gestão de releases determina quando um release será liberado para o cliente, gerencia o processo de criação e de meios de distribuição, bem como documenta o release.

Um release pode incluir:

- **Arquivos de configuração**
- **Arquivos de dados**
- **Um programa de instalação**
- **Documentação eletrônica ou em papel**
- **Empacotamento e publicidade associada**

1. **Solicitação de alteração**
2. **Geração de relatório de avaliação**
3. **Aprovação**
4. **Sim**
5. **Gera a ECO e Realiza a alteração**
6. **Realiza garantia da qualidade**
7. **Distribui a nova revisão**

Figura 15 – O processo de controle de alterações.

MANUTENÇÃO

Realizada a implantação do software com sucesso, iniciamos imediatamente a próxima etapa: a manutenção.

A etapa mais longa do ciclo de vida do software, a manutenção inclui a correção de defeitos não identificados nas etapas anteriores do processo de desenvolvimento de software, no aprimoramento da implementação dos subsistemas e na disponibilização de novas funcionalidades a partir da identificação de novos requisitos.

Pouco tempo depois de iniciada a etapa de manutenção, começam a chegar aos engenheiros de software os relatos de erros, bem como solicitações de adaptações e melhorias. Estas devem ser planejadas, programadas e executadas. Reiteramos aqui a importância do processo de gestão de alterações descrito anteriormente e ilustrado na Figura 15.

Outro problema associado à manutenção é a mobilidade dos engenheiros de software, pois, com o passar do tempo, a equipe que trabalhou no desenvolvimento do software pode ter sido substituída ou mesmo dissolvida, ou seja, pode ocorrer a inexistência de um engenheiro de software que conheça o sistema.



Neste ponto, podemos destacar a importância do correto cumprimento de todas as etapas do processo de desenvolvimento de software, de modo a gerar uma solução bem estruturada para o projeto, incluindo os diversos modelos que representam decisões tomadas ao longo do projeto, o que permite o entendimento da solução por parte de um engenheiro de software que não trabalhou no desenvolvimento do produto.

Na verdade, toda solução que aplica as melhores práticas da engenharia de software busca o cumprimento do requisito não funcional denominado manutenibilidade, sendo esse um indicativo qualitativo da facilidade de corrigir, adaptar ou melhorar o software.

Importante! Um dos grandes objetivos da engenharia de software é criar sistemas que apresentem alta manutenibilidade.

MANUTENÇÃO *VERSUS* REENGENHARIA DE SOFTWARE

Vamos considerar determinado produto de software implantado. Ao longo do tempo, a equipe de engenheiros de software realiza as manutenções necessárias em função de erros detectados ou de alterações geradas a partir da volatilidade dos requisitos. Paralelamente, a tecnologia vai se tornando obsoleta, e a manutenção, cada vez mais difícil.

O que você faria? Que tal reconstruirmos o produto com mais funcionalidades, melhor desempenho, confiabilidade e manutenibilidade? Esse processo é denominado reengenharia.

A reengenharia possui dois níveis: reengenharia de processos de negócio e reengenharia de software.

No primeiro nível, também denominado estratégico, a reengenharia de processos de negócio visa – por meio de alterações em regras de negócio, ou seja, das tarefas que permitem a obtenção de um resultado de negócio – o aumento da eficiência e competitividade de uma empresa.

Destacamos que um software tem como importante objetivo agregar valor a uma organização por meio da automação de processos; portanto, a referida alteração no negócio deve ser absorvida pela engenharia de software, gerando dois cenários:

Cenário 1

Que exige a construção de um novo software.

Cenário 2

Que permite a modificação de um software existente.

No segundo cenário, aplicamos a reengenharia de software, o que extrapola os objetivos da etapa manutenção.

Resumindo

Neste módulo, destacamos a importância das etapas de implantação e manutenção do processo de desenvolvimento de software.

A implantação inclui a migração do software de um ambiente de desenvolvimento para um ambiente de produção, permitindo a utilização do software pelos diversos usuários envolvidos.

Após a etapa de implantação, temos a manutenção. A existência de erros descobertos em produção é um fato; para tal, a equipe de engenheiros de software terá de aplicar as tarefas definidas no processo de gerenciamento de alterações. Outro motivo comum de solicitação de manutenção ocorre em função da volatilidade dos requisitos.

Com o tempo, as solicitações vão sendo intensificadas em função de alterações de regras de negócios ou pela obsolescência da tecnologia, podendo-se, neste caso, aplicar a reengenharia de software, o que extrapola a etapa de manutenção.