

Tópicos 1:

(C)

HIGOR AMARILHO DE SOUZA ①
jul

Os processos ágeis de software são modelos de processos que priorizam o desenvolvimento de software em etapas curtas e iterativas, entregando valor em ciclos contínuos que em geral podem variar entre duas e quatro semanas.

Os processos ágeis surgiram como alternativa aos processos prescritivos de desenvolvimento, nos quais o sistema é, de forma geral, entregue após todo o levantamento de requisitos, documentação, planejamento, design, codificação e testes. A construção de software usando processos prescritivos torna difícil, por exemplo, a mudança dos requisitos levantados no início do projeto de um sistema, já que geralmente é feita somente quando o sistema é entregue. Exemplos de processos prescritivos são o Modelo em Cascata, variações como o Modelo em V e o Modelo em W, assim como o Modelo em Espiral. Este último introduziu a ideia de ciclos iterativos de desenvolvimento, em que protótipos do sistema são construídos ainda nas fases iniciais como forma de evitar o problema da validação tardia por parte do cliente.

Os processos (ou metodologias) ágeis priorizam pensar e o software (produto) ao invés de processos e documentação. Para ter valor o produto deve ser entregue em iterações curtas para o cliente, de forma que as funcionalidades básicas mais importantes podem começar a ser validadas e usadas pelos clientes. Esse contato prévio do cliente com o sistema colabora o melhor entendimento sobre o produto que está sendo desenvolvido e evita que a validação do cliente ocorra somente ao final de todo o processo.

Além dos ciclos curtos, os processos ágeis seguem vários princípios que visam simplificar as etapas de desenvolvimento e agregar valor ao produto. Alguns desses princípios são:

- Entrega de valor: os ciclos curtos de entrega de funcionalidades colabora para que os clientes percebam a utilidade do sistema em desenvolvimentos. Além disso, eventuais divergências no entendimento das funcionalidades podem ser resolvidos logo.

- Documentação mínima: somente os documentos necessários são elaborados quando houver necessidade. Um dos documentos mais usados são as histórias do usuário, que descrevem as situações e objetivos do cliente no uso do sistema.

- Comunicação: a equipe de desenvolvimento realiza reuniões curtas para priorizar as tarefas diárias e de cada iteração, relatar avanços e comunicar problemas.

- Participação do cliente: em processos ágeis os clientes estão presentes durante todo o processo de desenvolvimento, esclarecendo dúvidas pontuais e propondo novas funcionalidades.

- Testes contínuos: cada nova funcionalidade é testada para garantir que as implementações estejam corretas (dentro do possível). Em alguns casos usa-se TDD (Desenvolvimento Dirigido por Testes). Em outros usa-se revisões por pares, de forma que as tarefas de desenvolvimento e testes de uma mesma funcionalidade sejam feitas por pessoas diferentes da equipe, o que ajuda a disseminar o conhecimento do código para toda a equipe e evita o encasnelamento dos testes por quem implementar uma dada funcionalidade.

Nos anos mais recentes, as metodologias ágeis têm sido cada vez mais adotadas por empresas de desenvolvimento de software, principalmente devido às características descritas anteriormente. Em geral, é difícil fazer avaliações precisas sobre os impactos do uso de processos ágeis sobre processos tradicionais. As avaliações feitas por estudos científicos geralmente usam técnicas de análise qualitativa, por meio do uso de surveys, questionários, entrevistas ou pesquisa-ação. Assim, a percepção dos impactos do uso de processos ágeis é muito mais prática

do que teórica.

Um exemplo de metodologia ágil é o Scrum. O Scrum ~~ff~~ caracteriza-se por ser um processo de desenvolvimento realizado por equipes com entre seis e dez pessoas. Existem alguns ~~papeis~~ ^{papeis} dentro da equipe, como o do Scrum Master, que geralmente é um membro da equipe com maior experiência e que atua como um facilitador (ou mentor). Há também o Product Owner, que é responsável por definir quais serão os produtos (funcionalidades) a serem entregues nos próximos sprints.

As iterações de entrega são conhecidas como Sprints, e costumam durar entre duas e quatro semanas. Os demais membros da equipe são desenvolvedores e designers, e toda a equipe forma o Scrum Team, sendo todos responsáveis pelo desenvolvimento do produto a ser entregue. No início de cada Sprint a equipe discute quais serão as próximas funcionalidades a serem implementadas e como elas serão divididas entre a equipe. Para isso, baseiam-se nos requisitos do usuário, estimando o tempo necessário para a realização das atividades de desenvolvimento. Além disso, o time realiza rápidas reuniões diárias (daily scrums) para atualizações sobre avanços e problemas encontrados, assim como as priorizações do dia.

Todas as atividades realizadas e a realizar são monitoradas através do uso de quadros com etiquetas (por exemplo, o Kanban), ficando em local visível e de fácil acesso para toda a equipe, o que colabora para a organização e a comunicação do sistema em desenvolvimento.

Tópico 3:

11/11/2024

Casos de uso são exemplos de utilizações de um sistema elaborados durante a fase de levantamento de requisitos.

Os casos de uso representam requisitos funcionais, ou seja, atividades que um sistema deve automatizar e disponibilizar para os seus usuários. Podem também representar requisitos não-funcionais.

A criação dos casos de uso começa após o levantamento inicial de quais são as funcionalidades (elicitações). O engenheiro de requisitos pode então detalhar tais funcionalidades em passos a serem feitos para atingir os objetivos de cada caso de uso.

O caso de uso pode estar relacionado com um ou mais perfis de usuário do sistema, que devem estar representados no documento correspondente. Além disso, casos de uso podem estar relacionados entre si, sendo, por exemplo, um caso de uso dependente de outro. Tais relacionamentos também devem estar representados.

Os casos de uso podem ser representados através de linguagem natural, ou através de diagramas. Nos casos de uso em linguagem natural, usa-se texto estruturado em seções, tais como objetivo, etapas, usuários e relacionamentos. O texto deve ser claro e preciso, evitando possíveis ambiguidades e imprecisões. A principal vantagem é que a representação em linguagem natural facilita a compreensão por parte do cliente.

Pode-se também representar os casos de uso através de modelos, tais como diagramas UML de caso de uso, diagramas de atividade e de sequência, que facilitam a compreensão por parte dos desenvolvedores.

Os casos de uso servem para elaboração dos requisitos e funcionalidades que o sistema deve ter e, assim como demais documentos, estabelece uma forma de explicitar o que deve ser feito e serve como referência para a implementação do sistema.

Exemplo de caso de uso

5

Considere um sistema de contabilidade no qual um usuário contador deseja fazer login no sistema. Para isso, ele deve digitar seu nome de usuário e senha.

Caso de uso

Objetivo: Entrar no sistema.

Usuário: Contador

Fluxo:

1. Usuário digita seu nome na tela de entrada do sistema.
2. Usuário digita sua senha na tela de entrada do sistema.
3. Sistema verifica o nome e a senha do usuário.
4. Usuário acessa a tela inicial do sistema.

Fluxo alternativo:

1. Caso o nome ou a senha do usuário não correspondam a nenhum usuário cadastrado.
2. Sistema avisa o usuário que o nome do usuário ou a senha são inválidos.

Tópico 9;

6

A Contagem de Pontos de Funções é uma métrica de software que visa estimar o esforço necessário para a construção de um software a partir das suas funcionalidades previstas. Cada função pode lidar com diferentes níveis de informações, como receber informações externas, fornecer informações internas ou externas. Cada fonte de informações tem um peso diferente na hora de calcular os pontos de função. Com isso, a Contagem de Pontos de Funções possibilita calcular o tempo estimado de criação de um sistema, o que por sua vez permite calcular o valor de custo do sistema de acordo com a distribuição dos horas trabalhadas da equipe de desenvolvimento.

A Contagem de Pontos de Funções é uma métrica que pode ser usada para medir custos ainda durante o levantamento de requisitos, já que não depende da implementação de código nem da linguagem de programação que será usada.

Outra característica da Contagem de Pontos de Funções é que ela tenta balancear a quantidade de horas estimadas com base em estimativas pessimistas, otimistas e realistas, dando um peso maior para as estimativas realistas. Esse tipo de métrica é mais adequado ao uso em modelos de processo prescritivos, nos quais os requisitos do sistema já estão bem definidos.