

Ferramentas CASE: Conhecendo algumas boas opções

Este artigo analisará várias ferramentas CASE que auxiliam a engenharia de software, desde o planejamento até a inspeção, ou seja, cobrindo todo o ciclo do software.

Fique por dentro

Este artigo é útil para desenvolvedores em busca de ferramentas para auxílio nas etapas de criação de software, esclarecendo as funções de cada plataforma e ferramentas abordadas, apresentando ferramentas e plataformas mais utilizadas na construção de um software. Com a leitura do artigo, o leitor poderá ter uma melhor visão, avaliação e julgamento quanto às ferramentas abordadas satisfazerem ou não suas necessidades.

No fim da década de 60 ocorreu a denominada crise do software. Resumidamente, a demanda e necessidade por novos softwares em conjunto com a dificuldade na definição de processos para criá-los acarretou em um importante evento realizado na Alemanha, que mais tarde veio a ser o marco da engenharia de software, a Conferência de OTAN. Esta que instituiu novas práticas para o desenvolvimento de software.

A [Engenharia de Software](#) versa sobre a qualidade do software aglomerada ao baixo custo e facilidade, buscando o aumento da produtividade nos processos de desenvolvimento. Simplificadamente, seu objetivo é o desenvolvimento, gerenciamento e documentação do software. Bauer, em 1969, definiu a Engenharia de Software como sendo o estabelecimento e uso de sólidos princípios de engenharia para obter software economicamente confiável e que trabalhe de forma eficiente em máquinas reais.

Ela aborda tanto questões relacionadas ao ambiente de desenvolvimento do software, como métodos, processos e ferramentas para sua produção. Deve-se ter bastante cuidado na criação de um software, pois quando há um mau planejamento e desenvolvimento, os recursos a serem gastos para conserto e

principalmente o tempo a ser disponibilizado para tal procedimento são aspectos inviáveis em qualquer mercado, sobretudo no tecnológico, onde essas duas características são indispensáveis.

O enfoque da Engenharia de Software é a análise e divisão do problema em pequenas partes, para que se possa realizar uma solução para cada uma delas e, depois de solucionadas, seja realizada a integração das mesmas. Para todo este controle, uma série de etapas devem ser seguidas, denominadas ciclo de um software. E o que vem a ser este ciclo? Basicamente, ele gira em torno de todas as fases determinadas de um software, desde o planejamento até o término, com o objetivo de cumprir todos, ou pelo menos quase todos os requisitos idealizados. Por ser um ciclo, as etapas não podem ser atropeladas sem antes se obter a verificação da qualidade em cada uma delas.

Para cada verificação de etapa existem ferramentas disponibilizadas na web; e com tantas tecnologias disponíveis para tornar o trabalho mais fácil, o próprio desenvolvedor tem dúvidas sobre qual delas escolher e o porquê de algumas serem consideradas melhores que outras.

Este artigo analisará várias ferramentas que auxiliam a engenharia de software, desde o planejamento até a inspeção, ou seja, cobrindo todo o ciclo do software. Todos os programas apresentados possuem a vantagem de serem livres e atenderem a uma gama de necessidades nas organizações. Claro que tais ferramentas podem não atender a todas as necessidades, visto que cada organização estabelece padrões e possui necessidades específicas.

Neste artigo, o leitor encontrará ferramentas que atendem aos processos de desenvolvimento, modelagem do processo, execução do processo, gerência de configuração, testes e inspeção, respectivamente:

- Plataforma Eclipse;
- Software BizAgi;
- Aplicação Redmine;
- Sistema Subversion;
- Framework JUnit;
- Ferramenta Sonar.

O esforço exigido dos desenvolvedores para planejamento, gerenciamento, acompanhamento, bem como a qualidade e facilidade exigida pelos usuários trazem à tona cada vez mais a automação dos processos. E, para isso, as ferramentas abordadas darão ao leitor uma melhor direção, auxiliando e orientando suas escolhas a partir dos pontos abordados em cada uma delas.

Processo de desenvolvimento de software

Processo se refere a uma série de ações ou operações visando um resultado. O processo de desenvolvimento do software é o resultado obtido do planejamento, no qual são avaliados aspectos como a definição de quais atividades serão realizadas, quando serão realizadas e quais as pessoas irão efetivá-las, além de haver uma padronização na realização do desenvolvimento; com a análise de custos, prazos, métodos, métricas e, claro, as ferramentas apropriadas para que este desenvolvimento ocorra de forma simples e ágil. O processo de desenvolvimento envolve um conjunto de pessoas em busca de uma meta, no caso, a criação de um software. Para tanto, o próximo tópico abordará a plataforma Eclipse, responsável por extrair o que foi determinado no planejamento e realizar o desenvolvimento concreto do projeto.

Eclipse

O [Eclipse](#) é uma plataforma livre IDE (*Integrated Development Environment* ou Ambiente Integrado de Desenvolvimento) fundamentada em [Java](#), que possui um conjunto de ferramentas de desenvolvimento, incluindo o PDE (*Plug-in Development Environment*), plug-in utilizado para desenvolvimento de novas ferramentas na integração com o Eclipse. A capacidade do Eclipse em aceitar a instalação de plug-ins é um adicional para sua utilização e também um dos motivos por ele ser uma das mais utilizadas plataformas de desenvolvimento, oferecendo apoio aos desenvolvedores para a realização de diversas funcionalidades em uma só plataforma. Apesar de ser escrito em Java, ele suporta várias outras linguagens de programação, como C++/C, devendo-se apenas realizar a instalação de plug-ins adicionais para sua utilização.

A plataforma comporta a reestruturação do código para que seja feito um aperfeiçoamento. Este procedimento é chamado de refatoração. É importante ressaltar que essa refatoração não modifica a semântica do projeto, ela apenas o aprimora. A plataforma engloba facilidades que vão desde a rápida visualização de todos os arquivos contidos no projeto até o gerenciamento de trabalhos coletivos.

Porém, o Eclipse apresenta algumas desvantagens. Por exemplo, a diversidade de plug-ins oferecidos poderá confundir o desenvolvedor (aquele iniciante da plataforma) em suas utilidades, podendo ocasionar atrasos. Outra desvantagem considerável é em relação ao tamanho da plataforma (considerada pesada), fazendo com que o ambiente de desenvolvimento se torne lento e contribuindo para uma queda na produtividade de desenvolvimento.

Com o Eclipse, a agilidade vai ao encontro das correções de erro, que são claros e objetivos ao desenvolvedor. A essa ferramenta é dada o nome de *manufacturer*. Ela é a responsável por interpretar o relatório de erros. Este procedimento acontece na compilação do código que contém erro, mostrando exatamente a linha onde ele está. Desenvolver sem possuir uma ferramenta facilitadora não é impossível, porém, complicado, o que derruba qualquer tentativa de agilidade.

Entre todas as funcionalidades do Eclipse, as citadas o fazem uma das plataformas preferidas dos desenvolvedores. Veremos a seguir o processo de modelagem e, logo após, um dos softwares mais utilizados neste método, o BizAgi.

Modelagem do processo

A modelagem do processo é a fase em que são analisadas as etapas de cada procedimento, tendo a finalidade de melhorá-lo. Com a modelagem, é possível obter uma melhor visão do processo, visto que ele é dividido na menor forma possível e ao mesmo tempo fazendo com que os envolvidos tenham uma visualização total do processo, apresentando uma análise clara e sendo um dos motivos principais do sucesso nos processos estratégicos da organização, pois oferece uma melhor percepção dos objetivos a serem alcançados pela organização.

Além de ser utilizada com a finalidade de melhoramento dos processos, o processo auxilia em treinamentos, discussões e medições, o que acarreta em um nível de maturidade e experiência maior à organização. A modelagem do processo na maioria das vezes faz uso de diagramas como forma de documentação. São empregadas técnicas, métodos e ferramentas para análise, modelagem e publicação, tendo por finalidade a otimização e controle de processos. Um dos instrumentos mais utilizados para a efetivação deste processo é o BizAgi, abordado a seguir.

BizAgi

O [BizAgi](#) é um importante software para mapeamento de processos e também para automação destes. Permite ao usuário mapear, documentar e também compartilhar seus processos através de desenhos facilmente interpretados. É usada a [modelagem BPNM](#) (*Business Process Management Notation* ou Notação de Modelagem de Processos de Negócio), padrão mundial de mapeamento, o que facilita a leitura do processo.

O **BizAgi** é considerado uma boa ferramenta devido à clareza na apresentação dos processos do software, auxiliando a visão de seus pontos fortes e fracos, como os custos, a complexidade, processos redundantes ou a falta deles. Além de se ter uma visão clara de cada etapa, é possível saber exatamente de onde continuar quando algo der errado, aumentando o desempenho e agilidade do negócio. Com o **BizAgi**, é possível detalhar minuciosamente aquilo que se pretende, conseguindo dar uma visão clara no entendimento para todos os envolvidos.

Na **Figura 1** foi mapeado o ciclo de um software e desenhadas as etapas necessárias para a conclusão do mesmo através do software BizAgi.

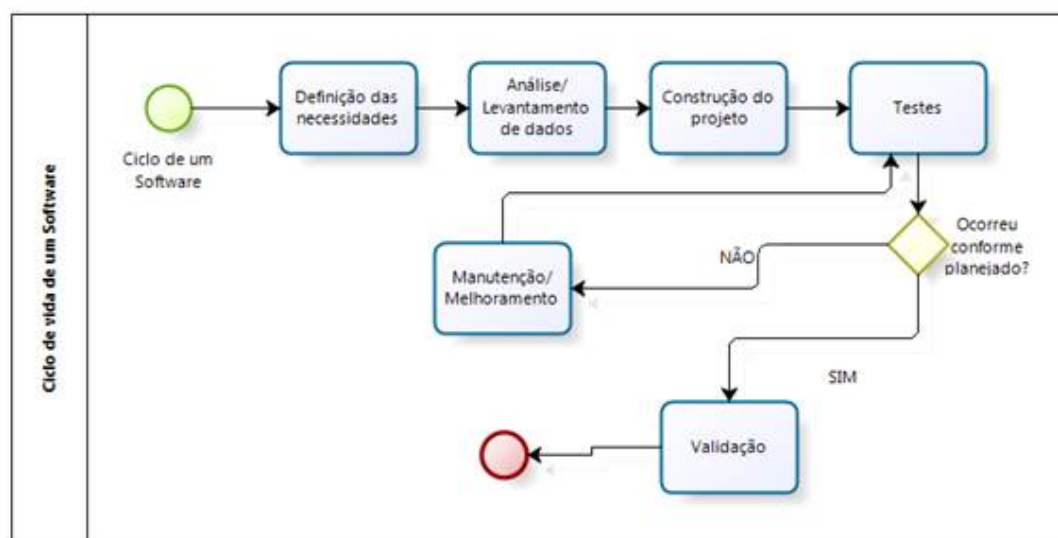


Figura 1. Fluxograma com o ciclo de vida de um software.

Com o mapeamento, é possível determinar tanto etapas que podem ser eliminadas, quanto aquelas que merecem mais dedicação e urgência, por exemplo. Com isso, é possível eliminar o desperdício de tempo e melhorar a qualidade dos processos.

Após todo o mapeamento do processo, se ocorrido conforme planejado, a próxima etapa é sua execução.

Execução do processo

O processo de execução está diretamente ligado ao processo de modelagem comentado anteriormente, pois aqui é colocado em prática aquilo que foi desenhado. Por isso a importância de uma modelagem completa e detalhada, tornando a execução o mais próximo possível daquilo que foi planejado e mapeado.

Portanto, o processo de execução nada mais é do que a efetivação do processo junto ao gerenciamento, permitindo a execução do projeto por etapas para que seja possível a realização de alterações, quando necessárias, e também estabelecer a garantia de seu funcionamento em conjunto. O processo de execução é considerado o mais delicado, pois é aqui que incidentes que não foram mapeados e previstos começam a surgir. Para tanto, ele é dividido em etapas, sendo: desenvolvimento do planejamento inicial, execução do que foi planejado e controle das mudanças necessárias.

Esse gerenciamento durante a execução do processo e, principalmente, a entrega do software nos prazos estimados tem sido uma das grandes dificuldades dos gestores, sendo um dos fatores determinantes para o sucesso ou fracasso do projeto. Para todas essas questões relatadas, entre outras, várias aplicações estão disponíveis, dentre elas, o Redmine.

Redmine

A aplicação Redmine é open source, sendo multiplataforma (programa ou sistema que pode ser executado em mais de uma plataforma, no caso do Redmine, é considerado multiplataforma por rodar em Windows, Ubuntu e as diversidades do Linux e no Centos e diversidades do Red Hat) e multibanco de dados (por suportar vários bancos de dados como, por exemplo, o MySQL e o PostgreSQL), voltado para gerência de projetos e escrita utilizando o framework Ruby on Rails (framework para desenvolvimento web) e de código aberto. Com o Redmine é possível dar suporte a vários projetos simultaneamente, ter um maior controle de acesso, configuração de controle por problemas, assim como receber notificações por e-mail. Os módulos do Redmine são:

- **Visão geral:** Este módulo apresenta uma visão geral do projeto, contando com um campo que pode ser editado informando para que serve o projeto. Exibe também um resumo das tarefas e quem são os membros participantes;

- **Atividades:** Relata quais as últimas atividades feitas no projeto; sendo assim, os envolvidos podem acompanhar cada alteração realizada;
- **Planejamento:** Módulo para controle dos sprints, ou seja, as versões dos softwares. Nele, há uma linha informando o percentual de conclusão da versão. Nesta aba, há a possibilidade de inclusão da quantidade de dias remanescentes para a conclusão do projeto, ou até de tarefas relacionadas a ele. É possível atribuir um repositório para cada projeto, já que os mesmos são separados em árvores de diretórios;
- **Tarefa:** Listagens de tarefas cadastradas. O módulo de Planejamento irá exibir informações sobre tarefas associadas à determinada versão;
- **Nova tarefa:** Essa subdivisão de tarefa foi desenvolvida justamente pensando na agilidade do processo. Ao ser cadastrada uma nova tarefa, automaticamente ela irá para o módulo de Tarefa. O tipo de tarefa a ser criada é dividido entre Defeito, Funcionalidade, Suporte e Administrativa;
- **Gantt:** O diagrama de Gantt é um recurso gráfico utilizado para visualizar a passagem de tempos ao longo das etapas de um projeto;
- **Calendário:** Exibição das datas das versões, assim como cronograma para seu término;
- **Notícias:** As notícias adicionadas podem ser acompanhadas via RSS;
- **Documentos:** O cadastro de documentos pode ser necessário se, por exemplo, tratar-se de um software com um termo de criação mais formal. Há ainda a opção neste módulo para a inclusão do manual do usuário do software, na entrega final;
- **Wiki:** Pode-se inserir páginas relativas a requisitos, mudanças, entre outros aspectos do software. Formulários podem ser criados dentro da própria aplicação, não necessitando de documentos paralelos para tal finalidade;
- **Arquivos:** Pode-se inserir qualquer tipo de arquivo desejado, que estará disponível sempre que necessário;
- **Configurações:** Configurações básicas do projeto como título, descrição, módulos a serem utilizados, pessoas envolvidas, entre outras.

O Redmine oferece a possibilidade de configurar uma tarefa de acordo com sua urgência. Por exemplo, se há um problema crítico em um software, é possível classificá-lo com os estados de prioridade alta, urgente ou imediata. Já quando o incidente é mediano, há a escolha entre prioridade baixa ou normal. Essa classificação permite uma melhor visão quando há vários incidentes a serem tratados, como ocorre na maioria das organizações.

Há também a opção de atribuir a tarefa a certo colaborador e adicionar observadores (aqueles envolvidos no projeto ou até mesmo para conhecimento de todos do departamento quanto à atividade a ser desenvolvida/melhorada/tratada) a esta tarefa.

Outra função é a determinação do tempo certo para cada tarefa, ou seja, prever o início e o fim. Além disso, é possível realizar a atualização no progresso do projeto por meio de porcentagem. Isso significa que, se no desenvolvimento do software algumas tarefas já foram feitas, o desenvolvedor poderá marcá-lo conforme a porcentagem de trabalho concluído. Os estados no Redmine variam entre Aberto, Em andamento e Fechado.

Podemos também anexar arquivos às tarefas, por exemplo, um tutorial do software ou qualquer outro arquivo que se achar necessário para um entendimento melhor da atividade.

Há inúmeras funcionalidades do Redmine para auxiliar qualquer tipo de projeto, além de plug-ins que podem ser adicionados e também a configuração de controle de acesso de acordo com o perfil do usuário, ajudando os gerentes de projeto tanto na parte de controle quanto na aprendizagem dos observadores. Por outro lado, um dos maiores problemas do Redmine é quanto a sua instalação e configuração, consideradas complicadas e cansativas. Há reclamações também devido à ferramenta não possuir um gerenciamento de riscos.

Além de todas as funções citadas, o Redmine permite a integração com vários repositórios, como o Subversion, software abordado a seguir.

Gerência de configuração

A gerência de configuração é o conjunto de atividades projetadas para controlar as mudanças pela identificação dos produtos de trabalho que serão alterados, estabelecendo um relacionamento entre eles,

definindo o mecanismo para o gerenciamento de diferentes versões destes produtos, controlando as mudanças impostas, e auditando e relatando as mudanças realizadas.

Esta etapa do processo é responsável por dar apoio ao desenvolvimento, pois as mudanças que surgem durante a etapa de desenvolvimento; as necessidades e requisitos tanto dos usuários quanto dos próprios desenvolvedores mudam a todo instante. A gerência de configuração não tem por objetivo impedir as modificações, e sim fazer com que elas sejam realizadas com o menor número de falhas. Para isso é importante a utilização de uma ferramenta que controle essas mudanças e ao mesmo tempo permita o equilíbrio durante os processos.

Subversion

Subversion ou simplesmente SVN é um sistema gratuito gerenciador de arquivos e diretórios, controlando todas as mudanças realizadas, permitindo que se tenha um maior controle de dados que foram adicionados/alterados e até removidos. Tem o objetivo de administrar versões diferentes de desenvolvimento de documentos. O sistema também é denominado de “máquina do tempo”, pois é possível realizar a recuperação de versões antigas dos dados.

Importante aliado, o Subversion tem uma excelente performance em rede, permitindo a utilização por pessoas em computadores diferentes, podendo ser utilizado para gerenciamento de qualquer diretório de arquivos.

Imagine a possibilidade de duas pessoas poderem trabalhar no mesmo projeto, ao mesmo tempo, até mesmo em lugares diferentes, sem que isso possa trazer nenhum problema de sincronização ou versão entre esses trabalhos. É isso que o SVN permite que aconteça.

O Subversion é bem utilizado na engenharia de software pela possibilidade de controlar diferentes versões de um projeto. Logo, todo o histórico, documentação, desenvolvimento e código-fonte de um software podem ser gerenciados através desse sistema.

Você deve estar se perguntando qual a utilidade do Subversion em sua organização. Bem, se há a necessidade de manter versões antigas com o objetivo de, por vezes, examiná-las e compará-las com as versões recentes, verificar mudanças e trajetórias dessas mudanças e o passo-a-passo, o Subversion é uma boa opção para este fim. Para controles de documentações ou quem elaborou aquele tipo de

documentação, às vezes em que foi editado e qual foi especificamente a alteração/remoção feita, este tipo de controle é possível com o Subversion.

Com tantas etapas, documentações que mudam a qualquer instante, por vezes a instabilidade de uma versão nova e a necessidade de verificação dos detalhes alterados ou removidos, torna-se mais fácil e ágil uma ferramenta como esta, que já mantém todo o controle, bastando às pessoas envolvidas sempre registrarem na documentação aquilo que foi editado. Outra vantagem do Subversion é o fato de ser gratuito.

Entre outras desvantagens do sistema, podemos abordar o fato de ser realizada a replicação de dados para todos os colaboradores envolvidos. Então, se o seu espaço em disco não é muito grande ou o compartilhamento envolve fotos ou pacotes de softwares, por exemplo, já não é vantagem sua utilização devido à sobrecarga de arquivos. Para isso, existem outros softwares considerados mais simples e que não sobrecarregam tanto, exemplo disso é o Unísson e o Rsync.

Aos usuários do Git, software com a mesma finalidade do Subversion, é oferecida a possibilidade da integração com o Subversion. É possível utilizar todas as funcionalidades do Git e após isso integrá-lo, ou comumente, fazer um push com o Subversion, como se este estivesse instalado localmente.

Testes

Na íntegra, o processo de testes é o responsável por criar todas as situações possíveis de uso do usuário do software com a finalidade de encontrar erros. Ou seja, o objetivo é encontrar um erro que não foi descoberto a fim de minimizar a entrega do software com falhas. O ciclo de vida do teste envolve as fases de planejamento, preparação, especificação, execução e entrega. A fase de testes é uma das mais importantes, pois a confiabilidade da empresa está em jogo.

Na fase de planejamento é organizada a forma como serão feitos os testes. Na preparação, o testador separa todas as ferramentas necessárias para a efetivação dos testes. Com a especificação, um ciclo é feito entre elaboração e revisão dos testes. Na fase de execução, é realizada a própria efetivação dos testes planejados, assim como o registro do que resultado obtido. Já na fase de entrega, a documentação adquirida com todos os testes é entregue aos responsáveis pelo desenvolvimento do software.

O processo de teste pode ser apoiado por várias ferramentas. Dentre as ferramentas que há no mercado, este artigo abordará o JUnit.

JUnit

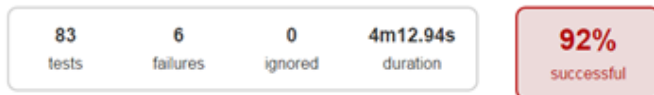
O JUnit é um importante framework criado com o objetivo de realizar testes automatizados em Java. Imagine, por exemplo, a construção de um carro, na qual pequenas partes são testadas e, após a verificação de que essas partes funcionam isoladamente, testes são feitos com as partes integradas. Isso facilita, não? É essa a proposta do JUnit, realizar testes de cada parte mínima do código isoladamente para que após isso seja feita sua integração.

Com esses pequenos parâmetros sendo testados isoladamente, há um aumento na qualidade do software. O desenvolvedor terá que realizar a verificação de onde o erro ocorreu, já que no JUnit é possível fazer esta verificação na menor parte do código do software e já ter a relação se aquelas pequenas partes estão em perfeito funcionamento ou não. O framework consegue analisar com rapidez os resultados dos testes e estabelecer métricas no ciclo do software.

Você deve estar se perguntando como é feita a classificação entre testes que deram certo e testes que falharam. É muito simples, o JUnit exibe uma porcentagem do quanto seu teste dará certo ou não. É exposta uma linha verde (para quando o teste estiver correto) e/ou uma linha vermelha (para quando der falha no teste), podendo seu teste estar 50% correto e 50% falho, e, caso der falha total ou parcial, o JUnit exibe onde está o erro.

Ao final dos testes, o JUnit exibe um relatório em formato HTML contendo informações detalhadas que foram obtidas dos testes realizados. Várias empresas possuem equipes próprias para testes, que já criam inúmeras possibilidades de testes, proporcionando uma agilidade maior ainda no desenvolvimento e, consequentemente, velocidade na criação e entrega do software. Veja um exemplo de relatório gerado pelo JUnit após a finalização de testes na **Figura 2**.

Test Summary



Failed tests Packages Classes

Class	Tests	Failures	Ignored	Duration
br.com.cybereagle.timedgoal.model.bean.DailyGoalTest	6	0	0	0.028s
br.com.cybereagle.timedgoal.model.bean.ExecutionTest	4	1	0	0.014s
br.com.cybereagle.timedgoal.model.bean.GoalTest	9	0	0	0.006s
br.com.cybereagle.timedgoal.model.dao.ExecutionDaoImpTest	5	5	0	4.248s
br.com.cybereagle.timedgoal.service.predictor.SweepLineServiceTest	1	0	0	4m7.09s
br.com.cybereagle.timedgoal.service.predictor.WeightEvaluationServiceTest	1	0	0	0.005s
br.com.cybereagle.timedgoal.service.scheduler.alarm.AlarmTaskSchedulerTest	3	0	0	1.409s
br.com.cybereagle.timedgoal.service.scheduler.executor.ExecutorTaskSchedulerTest	3	0	0	0.013s
br.com.cybereagle.timedgoal.service.timer.CountdownTimerImpTest	28	0	0	0.038s
br.com.cybereagle.timedgoal.service.timer.TimerImpTest	23	0	0	0.094s

Figura 2. Relatório gerado do JUnit.

Inspeção

O processo de inspeção busca a qualidade do software, ou seja, se ele está conforme as especificações estabelecidas inicialmente. Este procedimento pode ser realizado em todas as etapas do software, sendo considerado uma das melhores técnicas para a localização de defeitos. A inspeção gira em torno da qualidade do software, incidindo sobre várias divisões de etapas para que uma análise completa seja aplicada.

Qualquer inspeção é feita a partir de uma lista de requisitos. Esses requisitos são planejados antes mesmo da construção do código-fonte. As ferramentas mais indicadas para este procedimento dependem do tipo de inspeção a ser realizada, que variam entre softwares de gerenciamento de atividades, planilhas de cálculo, ferramentas de análise, entre outras. O importante mesmo são os resultados obtidos a partir dessas inspeções, que normalmente são feitas no código-fonte do software, mas não estão limitadas a isto, podendo abranger a especificação de requisitos, documentação do projeto, dentre outras.

Sonar

O software Sonar é uma ferramenta cujo objetivo é auxiliar na qualidade do desenvolvimento do software. Ele fornece um retorno ao desenvolvedor, proporcionando suporte à inspeção contínua do software.

Sonar é uma ferramenta livre que avalia a qualidade do software em todas as suas características: códigos duplicados ou sujeitos a possíveis bugs e até quanto à qualidade de sua arquitetura. A avaliação

feita pelo Sonar traz uma dinâmica que apresenta ao desenvolvedor um relatório bastante visual da porcentagem de qualidade do software. Existe uma opção bastante utilizada pelos usuários do Sonar, que é a geração de relatórios diários, ou seja, o desenvolvedor poderá programar o Sonar para exibir relatórios diariamente ou até mesmo a cada inserção ou modificação no código dentro do repositório. Dessa maneira, ele terá um melhor apoio para tomada de decisões a cada mudança feita durante o dia, o que garantirá qualidade na entrega de cada demanda.

A Engenharia de Software, assim como todas as áreas que envolvem a tecnologia da informação, vem adquirindo um espaço muito importante nas organizações, envolvendo uma gama de responsabilidades e se tornando fundamental para o bom andamento estratégico e organizacional das empresas.

O conjunto de ferramentas utilizadas pelo desenvolvedor não é o que define se uma empresa é ou não competente, mas sim o modo como essas ferramentas são utilizadas e se estão sendo empregadas da melhor forma de acordo com o padrão estabelecido pela própria organização.

Neste artigo foram vistas algumas ferramentas de apoio à engenharia de software, estas que cobrem todo o seu ciclo, desde a criação até a entrega. Ferramentas para controle e apoio aos desenvolvedores, cujos principais objetivos são a agilidade, praticidade e qualidade do produto.

Foram apresentados conceitos de cada ferramenta, e, logo após, os porquês de serem fundamentais nas etapas dos processos de desenvolvimento de um software.

Links

Eclipse Process Framework: Uma nova visão da Engenharia de Software

[https://fenix.tecnico.ulisboa.pt/downloadFile/3779571661065/
Eclipse%20Process%20Framework_IST.pdf](https://fenix.tecnico.ulisboa.pt/downloadFile/3779571661065/Eclipse%20Process%20Framework_IST.pdf)

Gerenciamento de Projetos com o Redmine

<http://cercomp.ufg.br/up/18/o/redmine-article.pdf>

