Aula 2º

Gerenciamento de mudanças

Olá, alunos(as),

Bem-vindos(as) a nossa segunda aula da disciplina "Gerência de Configuração". Nesta segunda aula, vamos começar a ver com mais detalhes as atividades da Gerência de Configuração. Nossos estudos serão divididos em três aulas. Na primeira aula, vamos ver uma base a respeito do Gerenciamento de Mudanças, que cuida de um aspecto muito importante de um projeto de software: As mudanças que podem ocorrer em nosso software.

Assim, não se esqueçam de ler esta aula, assistir aos nossos vídeos e fazer as atividades. Se tiverem alguma dúvida, vocês podem usar o nosso quadro de avisos, que eu responderei.

Bons estudos!

Objetivos de aprendizagem

Ao término desta aula, vocês serão capazes de:

- entender a importância do Gerenciamento de Mudanças;
- saber a importância da adoção do Gerenciamento de Mudanças;
- conhecer as principais características de um software de Gerência de Mudanças.

Seções de estudo

1 – Gerenciamento de mudanças

1 - Gerenciamento de mudanças

Para quem se recorda desde os primórdios de nosso curso, sabe muito bem que as mudanças são algo inevitável em nossos projetos de software, pois o software é algo em constante evolução, além do fato de seu ambiente, ao qual está inserido, também estar em constante evolução. Assim, é necessário que o software sofra mudanças para continuar sendo útil para os consumidores desse sistema.

Existem vários fatores que podem causar mudanças. Hélio Engholm Jr. nos dá alguns exemplos. Vamos ver?

- solicitação dos envolvidos no projeto;
- recomendação dos membros da equipe;
- atrasos em atividades do cronograma;
- necessidade de retrabalho;
- mudanças legais;
- necessidade de medidas corretivas/preventivas no projeto/
- evento externo, como evento legal ou concorrência;
- erro ou omissão da definição original do escopo do produto.
- erro ou omissão da definição do projeto;
- dificuldade com fornecedores;
- prazos, orçamentos, replanejamentos, equipe.
- escopo (ENGHOLM JR., 2010, p. 247).

Mas, nem toda mudança pode ser implantada. Os motivos podem variar, mas citamos alguns para ilustrar:

- A mudança aumenta os custos do projeto;
- A mudança introduz novos bugs no sistema;
- A mudança só beneficia um conjunto pequeno de usuários.

As mudanças, se forem adotadas de uma maneira sem controle e critério, podem causar problemas na evolução do sistema, prejudicando a credibilidade do programa. A seguir, mostramos dois casos em que as mudanças podem causar problemas no sistema e queda de credibilidade do produto.

Atualização do Windows 10 causa problemas no Menu Iniciar

A Microsoft lançou na quinta-feira (3) uma atualização cumulativa para usuários do Windows 10 e, adivinhe... o update deveria, entre outras coisas, corrigir uma revisão de bug de impressora — que havia acabado de ser introduzido —, mas acabou trazendo novos problemas, principalmente com o Menu Iniciar.

De acordo com relatos no fórum Reddit e da própria Microsoft, os usuários estão recebendo uma mensagem de erro crítica. "Se o menu Iniciar não está funcionando, tentaremos corrigi-lo na próxima vez que você fizer login", prometeu a Gigante de Redmond. Outra falha apresentada é a incapacidade do próprio release KB4524147 de completar o update. Várias pessoas vem dizendo que a instalação não consegue chegar até o fim em determinadas máquinas.

Como a Microsoft não tem nada específico sobre isso em sua página de suporte, parece que o número de consumidores afetados ainda é pequeno. Vale destacar que essa distribuição vai para mais de 900 milhões de usuários com milhares de configurações diferentes. A expectativa é de que a empresa lance mais um update em breve, justamente para ajustar o que ficou para trás desta vez.

Fonte: YUGE, Claudio. Atualização do Windows 10 causa problemas no Menu Iniciar. Canaltech, 2019. Disponível em: https://canaltech.com.br/windows/atualizacao-do-windows-10-causa-problemas-no-menu-iniciar-151667/. Acesso em 04 out. 2019.

Mais problemas! Xiaomi Mi A1 tem bateria sugada pelo Android Oreo

O histórico do Xiaomi Mi A1 com o Android Oreo está quase virando uma novela. Ela ganhou mais um capítulo, mas antes é válido relembrar a turbulenta situação do aparelho chinês:

12 de dezembro de 2017 – recebeu a atualização ao Android Oreo beta

25 de dezembro de 2017 – recebeu a atualização ao Android Oreo final

11 de janeiro de 2018 – atualização é suspensa por conta dos bugs

15 de janeiro de 2018 – atualização volta a ser liberada pela Xiaomi

25 de janeiro de 2018 – atualização volta a ser suspensa por conta dos bugs

Agora, os usuários que já atualizaram começaram a inundar o fórum da Xiaomi com reclamações sobre a bateria do Mi A1. De acordo com eles, o novo sistema está "sugando" a carga do telefone, reduzindo consideravelmente a autonomia.

Ademais, há outros pequenos erros mencionados:

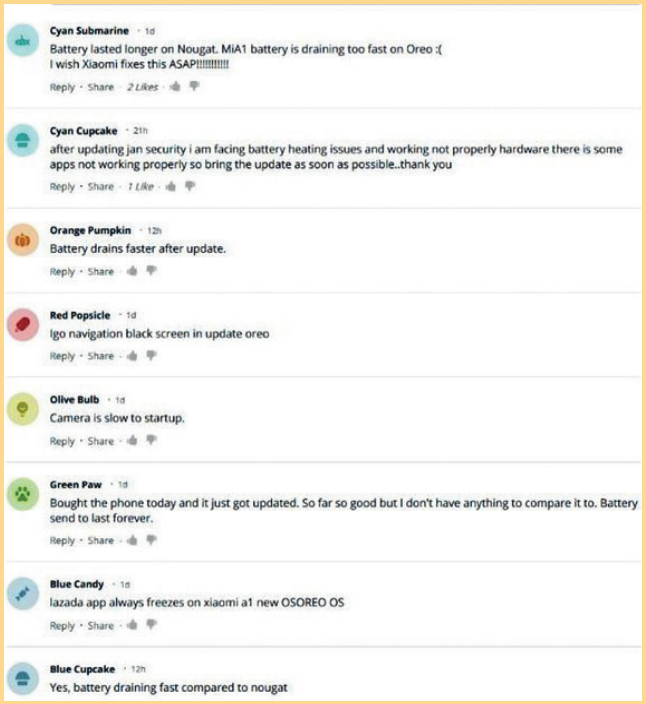


Imagem disponível em: https://t.tudocdn.net/316305?w=660&h=616.

Ainda no fórum oficial da Xiaomi, uma votação foi aberta para perguntar quem estava sofrendo com problemas na bateria do Mi A1. A maioria dos clientes, 74 para ser exato, confirmou estar com a autonomia reduzida, contra 15 pessoas que não identificaram o bug, veja:

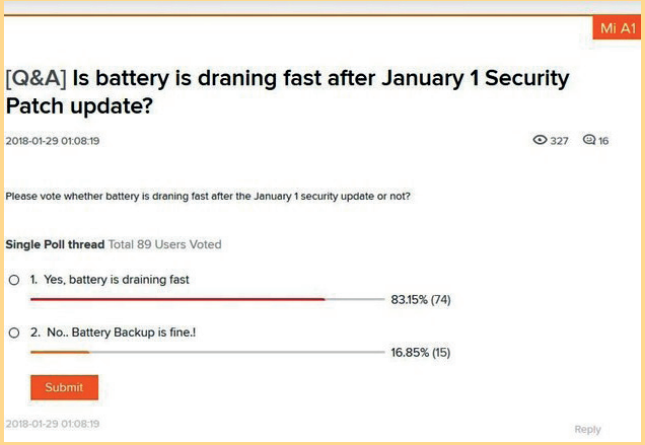


Imagem disponível em: https://t.tudocdn.net/316306?w=660&h=435.

Como consequência, Xiaomi ainda levará um tempo para voltar a liberar o Android Oreo para o Mi A1. É válido lembrar que o aparelho já está sendo vendido no Brasil, mesmo que de forma não oficial. (...)

Fonte: SANTOS, Leonardo. Mais problemas! Xiaomi Mi A1 tem bateria sugada pelo Android Oreo. Tudo Celular, 2018. Disponível em: https://www.tudocelular.com/android/noticias/n119100/Mais-problemas-Xiaomi-Mi-A1-tem-bateria-sugada.html/. Acesso em 04 out. 2019.

Assim, você viu aqui que é muito importante em um projeto de *software* ter as mudanças no projeto implantadas de maneira pensada e controlada, para evitar problemas e contratempos por parte das pessoas que vão utilizar o seu produto.

Assim, agora que você sabe a importância de adotar o gerenciamento de mudanças, vamos agora mostrar para você a definição do Gerenciamento de Mudanças, que é uma das quatro atividades que compõem a Gerência de Configuração. Sommerville (2011, p. 478) afirma que o Gerenciamento de Mudanças "destina-se a garantir que a evolução do sistema seja um processo gerenciado e que seja dada prioridade às mudanças mais urgentes e efetivas".

Já Hélio Engholm Jr. nos oferece outra definição, complementar a de Ian Sommerville. Vamos ver?

> Gerenciamento e controle de mudanças não consistem na prevenção de mudanças, mas na identificação e no gerenciamento de possíveis mudanças que possam vira a ocorrer no projeto, realizando-se uma análise de possíveis impactos no orçamento, cronograma, escopo e qualidade. O gerenciamento de mudanças garante o gerenciamento proativo das mudanças conforme elas ocorrem, além de garantir que serão gerenciadas durante todo o projeto. Desse modo, podemos dizer que a finalidade do gerenciamento de mudanças é assegurar que as mudanças em um projeto sejam consistentes e aprovadas pelos envolvidos no projeto e que estes sejam informados do estado do produto, das mudanças feitas e dos impactos gerados por essas mudanças em relação a custo e esforço. (ENGHOLM JR., 2010, p. 246).

Assim, em linhas gerais, podemos definir o Gerenciamento de Mudanças como os **processos e as ferramentas que apoiam o processo de controle e decisão da implantação das mudanças em um projeto de software**.

Para adotarmos o Gerenciamento de Mudanças, é muito importante definir um procedimento que deve ser utilizado quando é realizada uma solicitação de mudanças em um projeto de software, para decidir se essa mudança será implantada ou não:

> O processo de gerenciamento de mudanças está relacionado com a análise de custos e benefícios das mudanças propostas, a aprovação dessas mudanças que valem a pena e o acompanhamento dos componentes do sistema que foram alterados. (SOMMERVILLE, 2011, p. 478)

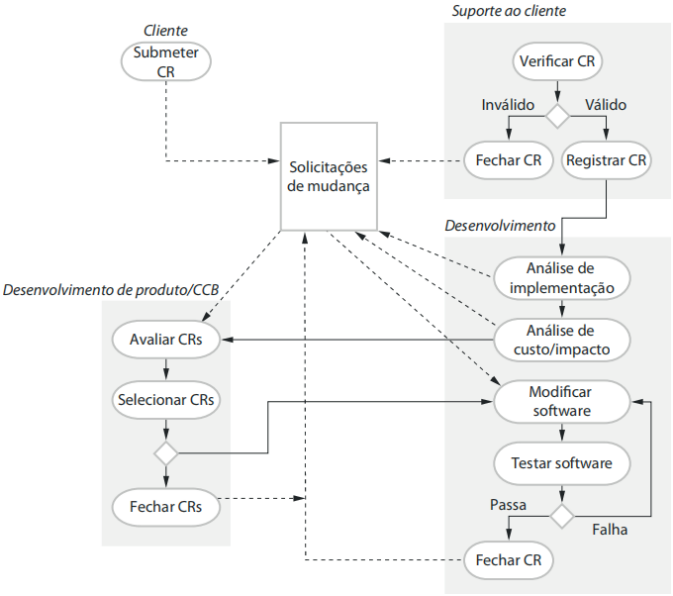
Assim, vários autores e normas sugerem *workflows* (processos) base para um processo de gerenciamento de mudanças no sistema. A seguir, vamos apresentar um desses processos, que é sugerido pelo renomado autor Ian Sommerville. Vamos ver?

1.1 – Processo para gerenciamento de mudanças

A seguir, vamos apresentar um modelo de processo para Gerenciamento de Mudanças de um projeto de software. Esse processo é sugerido no livro de Ian Sommerville, um dos mais respeitados estudiosos da área de Engenharia de Software. Outros autores podem sugerir outros modelos de processos, mas muitas das tarefas que os autores podem sugerir podem ser similares ao que será apresentado a seguir.

A figura a seguir representa um fluxograma, que apresenta graficamente o modelo do processo de gerenciamento de mudanças de Ian Sommerville. A seguir, vamos explicar como funciona cada etapa do processo.

Figura 1 – Modelo para um processo de Gerenciamento de Mudanças



Fonte: SOMMERVILLE, 2011, p. 478

Em suma, a ordem das etapas desse modelo de processo

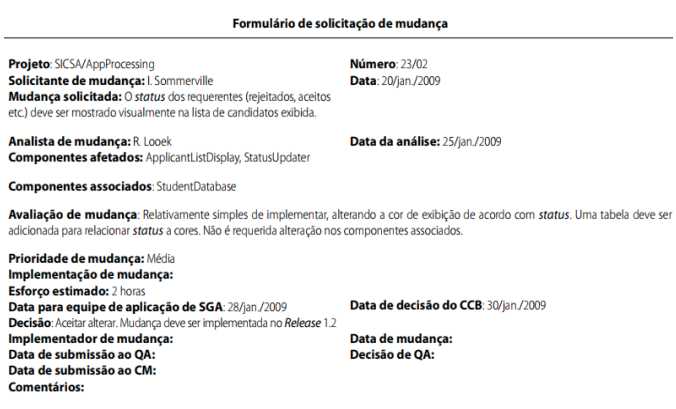
é a seguinte:

1. Cliente submete o formulário de solicitação de mudanças (CR – do inglês *Change Request*);
2. Uma equipe ou uma pessoa analisa se o pedido de mudança é válido. Se não for, o pedido é fechado. Se for válido, o pedido avança para a próxima etapa;
3. A equipe de desenvolvimento verifica a complexidade de implantação da mudança, analisando o custo e o impacto dessa mudança;
4. Com a análise feita, a equipe de desenvolvimento ou um Comitê de Controle de Mudanças (CCB – do inglês *Change Control Board*) avalia quais mudanças serão implantadas e fecha os pedidos que foram rejeitados;
5. A equipe de desenvolvimento implementa as mudanças aprovadas e testa se as mudanças causam problemas de regressão ou novos problemas no sistema. Se as mudanças forem testadas e aprovadas, a solicitação de mudança é fechada, indicando que a mudança foi feita.

Agora que você tem uma visão geral, vamos ver com um pouco mais de detalhes esse processo. Como você viu, o primeiro passo é quando o cliente (ou seu representante) preenche o formulário de solicitação de mudanças. Essa fase é denominada de **"Submeter CR"**. Nesse formulário, ele indica o seu nome, a data da solicitação e a descrição do que ele quer na mudança – correção de um bug, adição de novo recurso, alteração visual nos menus, entre outros. Nesse formulário, o cliente justifica o porquê que a mudança deve ser implementada no sistema.

Esse formulário pode ser manual ou eletrônico, onde podem ser registrados os resultados dos próximos passos da análise da implantação da mudança, que é iniciada após a submissão da mudança. A seguir, apresentamos um modelo preenchido de um formulário de solicitação de mudança hipotético, onde não são registrados somente os dados do pedido do cliente. São registrados também os dados das etapas posteriores da análise da mudança, que veremos a seguir.

Figura 2 – Modelo para um processo de Gerenciamento de Mudanças



Formulário de solicitação de mudança

Projeto: SICSA/AppProcessing
Número: 23/02
Solicitante de mudança: I. Sommerville
Data: 20/jan./2009
Mudança solicitada: O *status* dos requerentes (rejeitados, aceitos etc.) deve ser mostrado visualmente na lista de candidatos exibida.

Analista de mudança: R. Looek
Data da análise: 25/jan./2009
Componentes afetados: ApplicantListDisplay, StatusUpdater

Componentes associados: StudentDatabase

Avaliação de mudança: Relativamente simples de implementar, alterando a cor de exibição de acordo com ***status***. Uma tabela deve ser adicionada para relacionar ***status*** a cores. Não é requerida alteração nos componentes associados.

Prioridade de mudança: Média
Implementação de mudança:
Esforço estimado: 2 horas
Data para equipe de aplicação de SGA: 28/jan./2009
Data de decisão do CCB: 30/jan./2009
Decisão: Aceitar alterar. Mudança deve ser implementada no ***Release*** 1.2
Implementador de mudança:
Data de mudança:
Data de submissão ao QA:
Decisão de QA:
Data de submissão ao CM:
Comentários:

Fonte: SOMMERVILLE, 2011, p. 479

Em seguida, temos a análise de validade da solicitação de mudança (a etapa **"Verificar CR"**). Essa verificação pode ser feita por um desenvolvedor ou por uma equipe de suporte da aplicação. Segundo Sommerville, essa verificação é necessária existem determinadas situações que podem ocorrer, como descrito no excerto a seguir:

> A verificação é necessária porque nem todas as solicitações de mudança requerem ação. Se a solicitação de mudança é um relatório de bug, o bug pode já ter sido relatado. Às vezes, o que as pessoas acreditam ser problemas são os equívocos do que se espera do sistema. Em algumas ocasiões, as pessoas solicitam recursos que já foram implementados, mas que elas não conhecem. (SOMMERVILLE, 2011, p. 479)

Ainda sobre essa etapa, Hélio Engholm Júnior nos oferece algumas perguntas que podem ser feitas ao analisar uma solicitação de mudança no projeto:

- Quem solicitou a mudança tem alçada (competência) para solicitar mudanças?
- A mudança solicitada deve ser implementada?
- A mudança solicitada está relacionada ao escopo do projeto?
- Os benefícios da mudança aumentam ou diminuem as chances de término e sucesso do projeto? (Se o projeto estiver ainda incompleto)
- Quais são as justificativas da solicitação da mudança?
- A mudança traz benefícios para o projeto?
- Quem solicitou a mudança tem autoridade para tal?
- A mudança é realmente necessária?
- A mudança já ocorreu? (ENGHOLM JR., 2010, p. 247-8).

Após a análise preliminar, é dado o veredito se o pedido é válido ou não. Se não for válido, o pedido é fechado e não é implementado. Se for decretado como válido, é repassado para a equipe de desenvolvimento, que fará a análise mais profunda da mudança.

Com a mudança válida, a solicitação é levada a equipe de desenvolvedores, que analisa os detalhes técnicos da implantação dessa mudança, estuda quais serão os componentes afetados pela mudança (etapa **"Análise de Implementação"**) e qual será o custo de implantar essas mudanças (etapa **"Análise de custo/impacto"**). É nesse momento que a mudança é vista pelo olhar técnico. Os desenvolvedores verificam o que deve ser feito para cumprir a solicitação e quanto tempo será despendido para fazer as alterações. As constatações obtidas nessa fase são colocadas no formulário de solicitação de mudança, como pode ser visto novamente na Figura 2.

Em seguida, a solicitação passará por um momento crucial: será decidido se a mudança irá ser desenvolvida ou não (etapa **"Avaliar CRs"**). Essa decisão pode ser delegada a própria equipe de desenvolvedores ou a um comitê separado, denominado de Comitê de Controle de Mudanças (CCB). A delegação dependerá da complexidade da mudança pedida, do projeto a ser desenvolvido ou do ambiente onde o sistema está sendo construído.

Engholm Júnior nos dá uma definição a respeito desse comitê:

> Empresas estruturadas têm um comitê de controle de mudanças responsável pela orquestração do processo de gerenciamento de mudanças. (...). Participa da análise dos impactos da mudança e na aprovação e atribuição da mesma. Os gerentes de projeto podem assumir o papel em empresas que não o tem. (ENGHOLM JR., 2010, p. 248).

A análise é feita, tomando como subsídios a análise técnica, de custos e de tempo estimado feito pela equipe de desenvolvimento. Diante desses subsídios, existem fatores que podem impactar pela aprovação ou rejeição da mudança. É o que veremos a seguir:

Os fatores de fazer a mudança ou não

1. **As consequências de não fazer a mudança.** Ao avaliar uma solicitação de mudança, você deve considerar o que acontecerá se a mudança não for implementada. Se a mudança for associada a uma falha de sistema relatada, a gravidade da falha precisa ser levada em consideração. Se a falha de sistema causar a interrupção do sistema, isso é muito grave, e uma falha em fazer a mudança pode interromper a operação do sistema. Por outro lado, se a falha tem um efeito secundário, como cores incorretas em um display, então não é importante corrigir o problema rapidamente, de modo que a mudança deve ter uma prioridade baixa.
2. **Os benefícios da mudança**. A mudança é algo que beneficiará muitos usuários do sistema ou é uma proposta que beneficiará principalmente o proponente da mudança?
3. **O número de usuários afetados pela alteração**. Se apenas alguns usuários forem afetados, a mudança pode receber prioridade baixa. Na verdade, a mudança pode ser desaconselhada caso ela possa ter efeitos adversos sobre a maioria dos usuários de sistema.
4. **Os custos de se fazer a mudança**. Se fazer a mudança afetar muitos componentes de sistema (aumentando, portanto, as chances de introdução de novos bugs) e/ou levar muito tempo para ser implementada, então ela pode ser rejeitada, dados os elevados custos envolvidos.
5. **O ciclo de release de produto**. Se acaba de ser liberada uma nova versão do software para os clientes, pode fazer sentido atrasar a implementação da mudança até o próximo release planejado.

Fonte: SOMERVILLE, 2011, p. 480

Com a avaliação feita, chega o momento mais crucial da solicitação de mudança, pois é feita a decisão se a mudança será implementada ou não. A fase é denominada de **"Selecionar CRs"**, onde são selecionadas e priorizadas as mudanças a serem implementadas pela equipe de desenvolvimento. As mudanças que forem rejeitadas são fechadas. As mudanças aceitas são priorizadas e entregues à equipe de desenvolvimento.

Com as mudanças selecionadas e priorizadas, a equipe de desenvolvimento inicia o processo de implantação das mudanças, desenvolvido e testando as mudanças. Quando a implantação for encerrada e testada, o pedido é fechado.

Vale lembrar que isso vale quando os sistemas são customizados, ou seja, criados sob demanda para um cliente. Quando o sistema é um produto de prateleira, ou seja, é vendido para vários clientes, e não há uma customização específica, esse processo é ligeiramente diferente:

Gerenciamento de Mudanças para Softwares de Prateleira

O gerenciamento de mudanças para produtos de software (por exemplo, um produto de sistema CAD), ao invés de sistemas que são desenvolvidos especificamente para um determinado cliente, precisa ser tratado de forma ligeiramente diferente. Em produtos de software, o cliente não está diretamente envolvido nas decisões sobre a evolução de sistema, portanto, a relevância da mudança para o negócio do cliente não é um problema. As solicitações de mudança para esses produtos vêm da equipe de suporte do cliente, da equipe de marketing da empresa e dos próprios desenvolvedores. Esses pedidos podem refletir sugestões e feedback dos clientes ou análises do que é oferecido pelos produtos concorrentes.

A equipe de suporte do cliente pode enviar solicitações de mudança associadas com bugs descobertos e reportados pelos clientes após a liberação do software. Os clientes podem usar uma página da Web ou e-mail para reportar bugs. Em seguida, uma equipe de gerenciamento de bugs verifica se os relatórios de bug são válidos e converte-os em solicitações formais de mudança de sistema. A equipe de marketing reúne-se com os clientes e investiga produtos competitivos. Eles podem sugerir mudanças que devem ser incluídas para facilitar as vendas de uma nova versão de um sistema para clientes antigos e atuais. Os próprios desenvolvedores de sistema podem ter algumas boas ideias sobre os novos recursos que podem ser adicionados ao sistema.

Fonte: SOMMERVILLE, 2011, p. 480

Esse processo que vimos aqui pode ser implantada de maneira manual, usando fichas de papel ou usando planilhas e documentos para registro das mudanças. Mas, existem *softwares* que podem ser usados para apoiar esse processo.

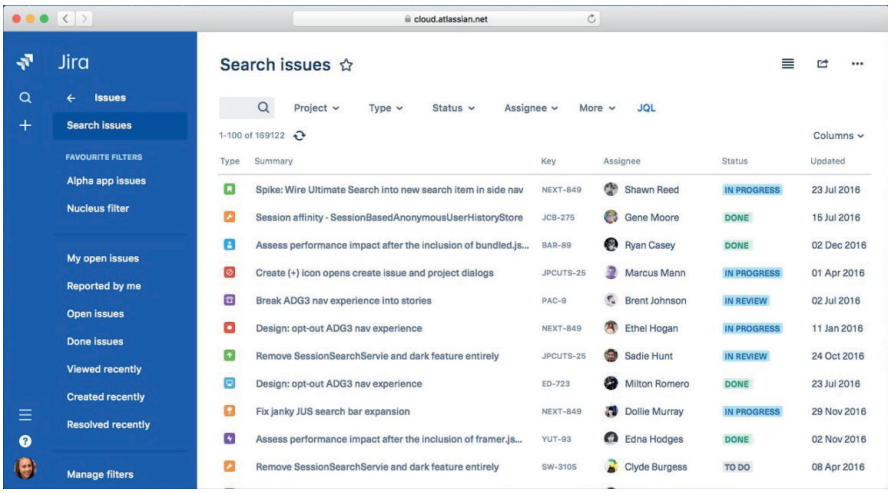
Um dos sistemas muito famosos é o Bugzilla, que é um programa de código-fonte aberto que oferece um sistema para registro das solicitações de mudanças, permitindo a discussão das mudanças, em um sistema similar a um fórum. Para instalar, é necessário ter um servidor Web configurado, um banco de dados e um interpretador para a linguagem Perl, linguagem a qual o programa foi escrito. Muitos sistemas utilizam essa ferramenta. O seu site oficial é https://www.bugzilla.org/.

Figura 3 – Screenshot da Plataforma Web oferecido pelo Filezilla

Fonte: Disponível em: https://commons.wikimedia.org/wiki/File:BugzillaScreenshot.png. Acesso em 27 out. 2019.

Outra ferramenta interessante é o Jira. Fornecido pela Atlassian, é uma plataforma Web paga que oferece uma plataforma customizada, sobre a qual os administradores têm total controle dos dados e dos usuários. Focado em projetos ágeis, é um dos sistemas mais utilizados em todo o mundo. Seu preço é definido pelo número de usuários que utilizam a plataforma, sendo que a empresa oferece um plano gratuito para até dez usuários, com recursos básicos para o gerenciamento. O seu site oficial é: https://www.atlassian.com/software/jira.

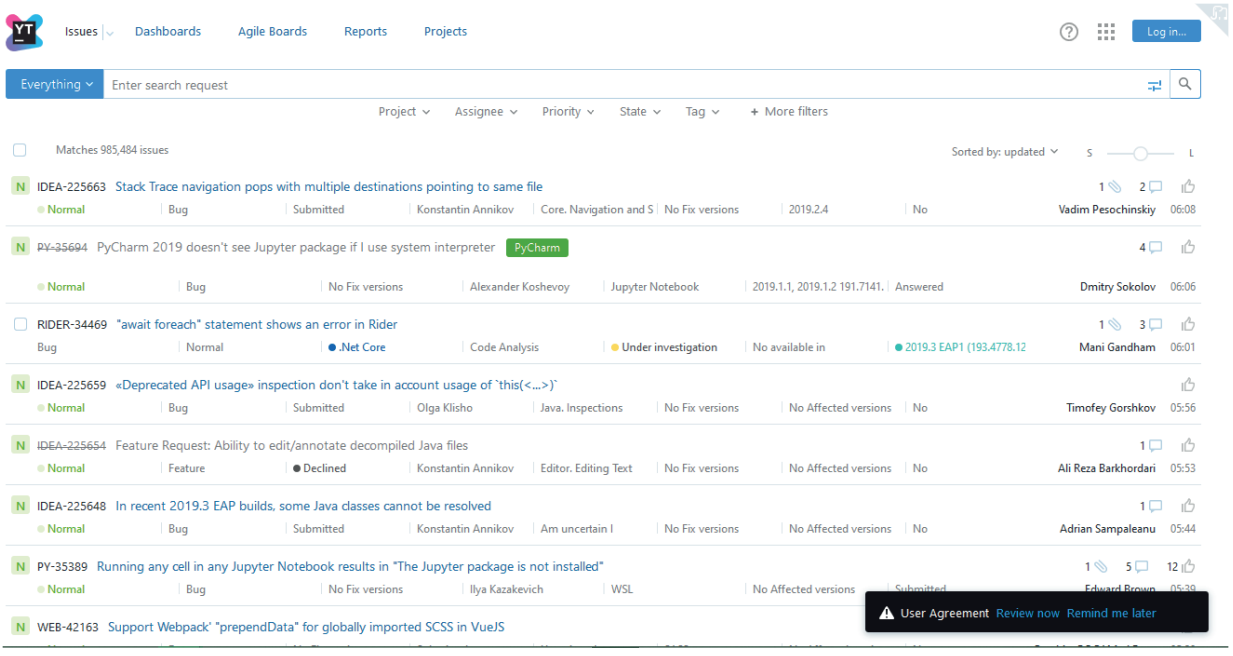
Figura 4 – Screenshot da Plataforma Web oferecido pelo Jira



Fonte: Disponível em: https://alternativeto.net/software/jira/. Acesso em 27 out. 2019.

Outro sistema interessante é o YouTrack, de propriedade da Jetbrains. Da mesma forma que o Jira, é uma plataforma web proprietária focada em gestão ágil. Seus recursos incluem quadros, estatísticas, relatórios e dashboards variados, que podem ser importantes para os gestores do projeto. Seu custo também é por usuário, sendo que equipes de até 3 pessoas podem usar gratuitamente. Seu site oficial é: https://www.jetbrains.com/youtrack/. Uma coisa importante que deve ser mencionada é que a própria Jetbrains usa esse sistema para o controle das solicitações de mudanças para as IDEs que a empresa desenvolve e vende. Esse sistema é público e disponível pela internet. O *link* de acesso para plataforma é: https://youtrack.jetbrains.com/issues.

Figura 5 – Screenshot da Plataforma Youtrack, para o gerenciamento de mudanças dos sistemas da Jetbrains.



Fonte: Disponível em: https://youtrack.jetbrains.com/issues/. Acesso em 27 out. 2019.

Agora que você sabe algumas ferramentas para o Gerenciamento de Mudanças, na próxima aula, vamos estudar sobre o Gerenciamento de Versões. Vamos lá?

Essa seção marca também o fim da nossa aula. Na próxima aula, vamos estudar a atividade de Gerência de Versões. Até lá!

Retomando a aula

Chegamos ao final da nossa segunda aula. Vamos relembrar?

1 – Gerenciamento de mudanças

Nessa seção, você viu que o Gerenciamento de Mudanças pode ser entendido como o conjunto de processos e as ferramentas que apoiam o processo de controle e decisão da implantação das mudanças em um projeto de software. Para que possamos entender isso, vimos um modelo para um processo de gerenciamento de mudanças e alguns exemplos de sistemas de controle de mudanças.

Vale a pena

Vale a pena **ler**,

ENGHOLM JR., Hélio. *Engenharia de software na prática*. São Paulo: Novatec, 2010.

SOMMERVILLE, Ian. *Engenharia de Software*. 9. ed. São Paulo: Pearson, 2011.

Minhas anotações