

Passagem de conhecimento entre equipes de projetos e equipes de operação

Paulo Marinho Leal
Analista de sistemas
paulo.marinho.leal@gmail.com

Resumo - Equipes de suporte necessitam do maior número possível de informação para atender e suprir da melhor forma as necessidades dos clientes, nesse ponto a documentação do sistema se faz fundamental. Processos, procedimentos e modo operante que devem ser passados pela equipe que concebeu o projeto. O problema é que nem sempre isso ocorre da forma como deveria, e as consequências dessa informação deficiente podem ser desastrosas. Técnicas ágeis de gestão de projetos priorizam o produto mais que a documentação desse produto, dando margem a má interpretação de que a documentação não é importante, ou que a mesma não deve ser realizada.

Palavras-chave: suporte, operação, base de conhecimento

Introdução

Em um mundo totalmente conectado, a necessidade de se manter atualizado é quase uma obrigação diária, fatos e acontecimentos que impactam a nossa vida ocorrem em qualquer lugar do mundo a todo o momento. Dentro deste contexto, os sistemas de informações que nos cercam necessitam acompanhar essa onda mudança cada vez mais rápida.

Para trazer a comodidade que esperamos, os sistemas necessitam estar em constante evolução, desta forma estarão sempre alinhados com nossas expectativas e necessidades diárias.

No centro desta evolução, estão as equipes de tecnologia da informação, que tem por missão evoluir e manter esses sistemas, que por sua vez estão divididas entre equipes de desenvolvimento e equipe de suporte e manutenção.

A primeira tem o objetivo de criar novas funcionalidades, e melhorar as funcionalidades existentes, a segunda está preocupado com funcionamento contínuo do sistema, busca contornar qualquer anomalia e mal funcionamento que possa apresentar, garantindo assim a principal função dos sistemas, trazer maior praticidade na vida humana contemporânea.

A fim de cumprir essa árdua tarefa, as equipes de suporte e operação necessitam conhecer esses sistemas, para isso uma base de informações sobre esse sistema é fundamental, informações passadas inicialmente pela equipe que concebeu o

sistema e ao longo do ciclo de vida do produto em operação, outras informações relevantes devem ser adicionadas para que a tenhamos sempre atualizada.

Contexto

Um projeto de software em sua grande maioria é composto de um produto de software, e uma documentação deste produto, seja ela abrangente ou mais resumida contendo informações relevantes sobre sua função, construção e modo de operação.

É justamente essa documentação tão necessária para conhecermos o produto, que vem se tornando cada vez mais escassa, por ser considerada secundária e tratada como desnecessária, é muitas vezes deixada para o fim do projeto com aquela frase típica “se sobrar tempo a gente faz”.

Como em projetos os prazos são cada vez mais desafiadores, quase nunca temos tempo sobrando, e conseqüentemente chegamos ao fim do projeto com alguns possíveis cenários relacionados a esta documentação:

- Produto com uma documentação superficial;
- Documentação que não reflete a realidade;
- Documentação que reflete parcialmente a realidade do produto ou incompleta;

Esses cenários podem acarretar em consequências que podem ser desastrosas, que vão desde a má utilização do produto pelo

cliente, até uma falha grave na operação onde se demore muito tempo para restabelecer o serviço.

Toda essa história começa no lado do projeto do produto de software, que por sua vez tem equipes de desenvolvimento, trabalhando hoje em sua grande maioria com metodologias ágeis. Metodologia essa que tem como principais pilares os seguintes valores:

- Indivíduos e interações mais que processos e ferramentas
- Software em funcionamento mais que documentação abrangente
- Colaboração com o cliente mais que negociação de contratos
- Responder a mudanças mais que seguir um plano

É em um dos pilares da metodologia ágil que está o cerne da questão “Software em funcionamento mais que documentação abrangente”, concordo que não há serventia alguma uma documentação abrangente ou resumida, se o produto não agrega valor ao negócio do cliente, que não funciona ou tem um funcionamento precário.

Por valorizarem a comunicação direta entre a equipe, cliente e até com os usuários, é natural que se documente menos, pois nesse contato define-se com mais facilidade os requisitos e as prioridades, reduzindo assim a necessidade de documentar. Aliada a outras práticas complementares muitos úteis e de forte adesão, como o clean code e o TDD.

Em contrapartida, após a entrega deste produto ao cliente, existirá uma equipe que terá a função de manter esse produto em operação, que tratará as demandas de melhoria/evolução deste produto, e as demandas provenientes dos incidentes tratados no dia a dia, devido as inconsistências do serviço, sejam elas bug, mal uso do produto entre outras.

Por isso, não podemos ter um sistema que pouco sabemos como funciona, quase como uma caixa preta, que quando há um problema as equipes de suporte tenham que perder tempo procurando interpretar como aquele sistema funciona, olhando direto no código, para só depois passar para a investigação do problema em questão. Nesse processo perde-se um tempo precioso, onde o serviço do cliente pode estar inoperante ou em mal funcionamento.

Para prestar o melhor atendimento principalmente em momento de crise, é necessário conhecer o produto em questão, e

nada melhor que uma base de conhecimento sobre esse produto, com detalhes sobre suas funcionalidades, requisitos, e os detalhes mais relevantes sobre implementação, para auxiliar as equipes de suporte possam manter o serviço em operação.

Existem diversas maneiras de documentar um software, onde em todas elas baseiam-se do princípio que a equipe de projetos ao fazer a transição para a equipe que irá manter o produto, faça a passagem de conhecimento e documente todo esse processo, e a partir desse momento caberá a equipe operacional a responsabilidade de manter atualizada essa base de conhecimento, que poderá ser alimentada em várias mãos, de diversas áreas, banco de dados, suporte, desenvolvedores, analistas de negócios, enfim, todos que tenham contato e que realizarem algo relevante em relação ao produto.

Existem diferentes visões sobre o mesmo produto com relação à base de conhecimento de sistema, uma foca na visão de negócio e a outra na visão de manutenção do mesmo.

A visão do negócio está preocupada em descrever as regras de negócios, os requisitos funcionais e as funcionalidades, relacionadas a cada área do produto, onde fica essas regras em cada tela. Isso é importante, claro para apoiarmos o cliente em uma dúvida, por exemplo quando ele liga para o help desk para perguntar como faz tal coisa no sistema. Geralmente, essa parte desta base de conhecimento será alimentada e atualizada por analistas de negócios, analistas de requisitos, atendentes do help desk, ou todos aqueles que não tenham a visão mais técnica.

A visão de manutenção, preocupa-se em especificar os requisitos mais técnicos, na maior parte dos casos é aquele que o cliente não vê, de como implementar as regras de negócios, como é a construção ou particularidade de uma tela por exemplo, que tecnologias usa, quais são os acessos, se tem log, como é o deployment do ambiente, onde está hospedado, entre outros. São informações de extrema importância para manter o sistema “rodando” e com o mínimo de interrupções. Esta é mantida geralmente pelas equipes de TI envolvidas nessa operação como por exemplo, analistas de suporte e de sistemas, DBAs, desenvolvedores, atendentes do help desk entre outros.

Para tal citarei algumas das práticas mais relevantes que podem ser utilizadas para construção dessa base de conhecimento, creio que todas podem ser utilizadas em conjunto:

User stories - Criadas em tempo de projeto, as user stories são o pontapé para a construção do sistema, é a partir delas que são pensadas as funcionalidades, que então são formadas as tarefas do projeto, é de vital importância que elas sejam incorporadas para a construção da base de conhecimento, pois ela faz parte da visão do negócio, onde poderemos dizer funcionalmente o que é o quê.

Descrições no controle de versão - A grande maioria das empresas, hoje fazem uso de um software de controle de versão do código fonte do projeto, que serve para gerenciar as diferentes versões de um software em tempo de desenvolvimento, nele se tem todo o histórico do que foi desenvolvido, e em cada commit realizado, é possível colocar um comentário descrevendo aquilo que foi feito. Esta descrição pode ser utilizada para alimentar a base de conhecimento do sistema, que no caso é a visão de manutenção, que esclarece como foi implementado tal regra de negócio, e etc.

Worklog de timesheet - Em projetos de software, é uma prática comum a utilização algum software para controle do tempo utilizado para desenvolver cada feature, onde o desenvolvedor aponta as horas trabalhadas na mesma. Em softwares como o Jira existe um campo de descrição, onde é possível descrever o que foi realizado naquele tempo. Assim como a descrição do controle de versão, podemos utilizá-la na base de conhecimento.

Wiki - Todas as práticas descritas acima não têm muito valor se não forem concentradas e organizadas em um único local, e nada melhor que a utilização de uma Wiki, onde podemos concentrar todo conhecimento do produto independente da visão. Nela é possível navegar entre os tópicos criados, nos menus, ou realizar uma busca para encontrar o conteúdo desejado. Possibilita criar conteúdo de diversos tipos, como as tradicionais descrições textuais, vídeos explicativos, tutoriais passo a passo, com prints com as telas do sistema entre outras.

Conclusões e Lições Aprendidas

O conteúdo apresentado neste artigo, teve o intuito de exibir a necessidade e o impacto que uma base de conhecimento ou a ausência dela, pode ter na operação do dia a dia em produtos de software.

Exemplificando a necessidade de se pensar na passagem de conhecimento do pós projeto, e que é possível com um pouco de planejamento e o auxílio de ferramentas simples construir uma base de conhecimento.

Sabemos que o artigo está longe de encerrar a discussão sobre o tema, que existem ainda muitos mitos em relação ao assunto.

Referências

1. MANIFESTO ÁGIL PARA DESENVOLVIMENTO DE SOFTWARE. Disponível em: <<http://www.manifestoagil.com.br/>>. Acesso em 25 abr. 2018.
2. MÉDIAWIKI. Disponível em: <<https://www.mediawiki.org/>>. Acessado em 02 de abr. 2018.
3. COUGO, Paulo Sérgio. ITIL Guia de Implantação. Rio de Janeiro: Campus editora, 2011.

Sobre os Autores:



Paulo Marinho Leal
Analista de Sistemas –
Working Minds Tecnologia
paulo.marinho.leal@gmail.com

Mini currículo

Graduado em Ciência da Computação pela Unicarioca, com pós graduação em Desenvolvimento Web e Gerenciamento de projetos pela PUC-MINAS, certificado PMI CAPM e ITIL Foundation, com 11 anos de experiência em tecnologia, 7 delas trabalhando com sistemas, atualmente é responsável pelo suporte e sustentação de sistemas na Working Minds Tecnologia da Informação.