Proteja-se ou Fuja: Coisas que fazem um projeto dar ruim

Salve padawan, em nosso artigo anterior apresentei minha visão sobre o termo Crise de Software, cunhado no século passado para explicar o porquê dos projetos informáticos falharem tanto.

Hoje vamos falar de situações que tendem a destruir com um projeto, detonando a equipe, deixando malucos líderes de projeto e exasperando clientes e usuários. Este texto tem o objetivo de apresentar situações e auxiliar a resolução dos problemas decorrentes.

O que significa a expressão: Um projeto dar ruim?

É muito ampla e genérica a resposta, mas em linhas gerais é falhar na entrega do Software, seja não cumprindo os requisitos, seja não respeitando o prazo acertado ou ainda pior estourando o valor orçado, encarecendo absurdamente o projeto.

Muitas vezes ocorre por um único erro, outras vezes acontece devido ao acumulo de pequenas situações errôneas, que acumuladas levam ao fracasso do projeto. Não se desespere, este não problema não é restrito a informática, ocorrendo em todas as áreas, onde uma parte contrata um serviço para outra parte.

Informática e o risco de software

Falar sobre o risco de software é mergulhar num ambiente de dúvidas, temores, confusões e fracassos. Naturalmente existe um conflito interno entre o Agente Comercial, que prospecta o mercado oferecendo soluções e vendendo serviços e a equipe técnica, que deverá entregar o produto ou serviço.

Uma cobertura de risco insuficiente leva um projeto ao fracasso com um ROI negativo e perdas de valor na imagem da empresa prestadora de serviço, cabe a alta administração trabalhar duro para manter a estrutura coesa e minimizar os riscos.

Lembrando que o cliente quer o máximo de retorno sobre o trabalho pago com o mínimo de custos, procurando a todo momento incluir mais solicitações e mais funcionalidades na ferramenta acordada.

O programador quer o maior salário com o mínimo de esforço possível.

E a empresa consultora ou prestadora de serviço: quer o maior retorno possível, visando prestigio no mercado e máximo retorno aos stakeholders.

Cabendo ao gestor minimizar o risco, protegendo-se nas partes frágeis e com a experiência adquirida, tentar prever futuros problemas e falhas.

Experiência com a crise de software e alguns fiascos.

Parece história de pescador, exagero do tiozão, mas acredito momentos insanos acontecem em projetos.

Problemas no Onboarding

Muitas vezes a infraestrutura do cliente falha redondamente a criar perfis de acesso ao ambiente de desenvolvimento, em alguns projetos chegaram a perder 15 dias de trabalho, pois o serviço era terceirizado e existia um longo ritual burocrático a ser cumprido, com inúmeros e-mails trocados e algumas horas de telefonemas até solucionar a situação.

Cisão de Sistemas

Algumas empresas ao abrirem uma sucursal em outro pais, penam na migração do software, problemas na localização e tradução das mensagens de erros, principalmente quando a empresa contratada faz um levantamento pífio e alega que diversos módulos estão fora do escopo.

Acidentes e baixas medicas.

Um evento altamente aleatório, mas que coloca a prova de toda a equipe, desde o falecimento de um membro importante a meio do projeto, a acidentes automobilísticos que retiram o profissional a meio do desenvolvimento.

Sendo o mais grave baixas psicológicas, onde uma analista com crise do pânico abandona o projeto a 30 dias da implantação, ocultando o serviço e nas reuniões de ponto da situação alegando que o serviço estava concluído, sendo que a verdade nem havia iniciado a codificar.

Insanidade do Gestor alocando gestante.

Esse foi realmente uma das loucuras mais loucas, no kick-off uma da principal analista estava em avançado estagio de gestação, colocada como recurso chave e calculado todos os prazos contando com sua experiência, porem antes de ser colocada a prova entrou de licença atrasando todos os planejamentos e sobrecarregando os membros restantes.

Chama mais um

No momento mais assustador do desenvolvimento, quando todos estão atolados de trabalho, sem tempo nem para ir ao banheiro, o gestor faz o onboarding de um novo recurso, na esperança de ajudar, acaba atrapalhando ainda mais, pois perde-se um tempo precioso, ensinando e introduzindo o novo recurso na equipe, bem como lutando contra a burocracia para criar novo acesso ao funcionário novo.

Sem lenço nem documento.

Quando se fala em onboarding, pensa-se apenas em criar user-id na rede, mas vai muito mais longe, com trabalho de liberação de acesso no edifício, criação de badge/cracha de acesso, permissões CRUD em pastas e arquivos da rede, apresentação a outras equipes, alocar espaço físico no escritório (tempos pre-pandemia), quem nunca trabalho, não imagina o enrrasco que é arrumar uma mesa e cadeira para trabalhar.

Ferramentas inadequadas de trabalho

Esse é o risco mais grave, quando o cliente obrigado o uso de determinada ferramenta, framework, lib e implementação que o mercado reconhece como obsoleta, problemática e de difícil implementação, mas mesmo assim temos que atender e usar. Reze para sair.

Leviandade na gestão dos riscos

Devido a imaturidade da equipe e falta de visão do Project Owner, muitas vezes os riscos são minimizados, ignorados e não se criam estratégias para proteger a equipe e o projeto dos inúmeros riscos inerentes ao desenvolvimento informático.

A equipe deve se unir para mapear os riscos e criar estratégias defensivas contra os danos e possíveis problemas, que possam surgir, lembrando que cada projeto é único, porem podemos adaptar e avaliar estratégias usadas em outros projetos para blindar e levar o projeto a bom porto.

Excesso de otimismo

Muitas vezes a solução é ruim, porem o ambiente de testes e aceite estão despovoados e a solução funciona lindamente com 10k a 50k registros, porem ao entrar em produção com seus 500k de registro, acumulado com a concorrência e o pior um crescimento exponencial, logo a solução quebra, as vezes nem saiu da garantia e esta desesperando a equipe de sustentação.

Escopo desconhecido e não sabido

Muitas vezes o Analista de Negócios não sabe bem o que o cliente quer, o analista comercial vendeu um mega produto e o usuário pediu a América, com isso na hora do desenvolvimento o caos se instala, pois existe 3 versões de um software esperado, mas ninguém sabe o que vai acontecer ou que software vai sair.

Horas extraordinárias a noite e finais de semana.

A pior parte de um projeto mal planejado, cheio de erros conceituais é a sobrecarga de trabalho, conforme o deadend se aproxima, o gestor descobre que as implementações não ficaram prontas a tempo hábil, com isso, começa-se a pressão sobre a equipe para vestir a camisa, isto é, dar um gás, em jornadas insanas de 10 a 12 horas e trabalhos pontuais nos finais de semana e feriado.

Receita de fracasso na certa, pois com a equipe cansada, começa a desmotivação e em alguns casos os profissionais começam a abandonar o barco, pois percebem que é uma causa perdida.

Todos ralham, mas ninguém tem razão.

Um desafio e tanto fazer a comunicação funcionar, imagine projetos internacionais com equipes em Portugal, Espanha, Inglaterra e Índia, problemas de fuso horário, problemas de conhecimento funcional e é claro problemas linguísticos.

Outras situações envolvem duas a três empresas prestadoras de serviço, cada uma com uma parte do projeto, respondendo ao usuário e comunicando-se entre si, tentando brilhar mais que as outras, puxando a brasa para suas sardinhas.

A comunicação impacta muito o bom andamento do projeto, dificultando e atrasando a boa execução do mesmo.

Excesso ou falta de gerenciamento

Muitas vezes o cliente deixa o projeto a deriva, o usuário sobrecarregado com sua dinâmica e dia a dia, esquece dos parceiros e deixa de acompanhar o andamento e atrasa-se com os retornos, deixando um vazio, que impacta o andamento do projeto.

Pior acontece quando o Project owner esta alocado em diversos projetos e acabando deixando o cliente no escuro, dando poucos feedbacks e posicionamentos a tempo e hora, neste caso, o usuário melindrado descera o porrete para solucionar este abandono.

Algumas vezes, mas raro é quando ocorre o excesso de comunicação e o usuário e ou Project owner perde-se em meio a tantos e-mails e mensagens.

Erro na estimativa.

Triste quando acontece, mas se a ferramenta de ponto de função for mal calibrada, ou as estimativas sofram de vieses otimistas o prazo final sofrera inúmeras alterações comprometendo a entrega, gerando atrasos e falhas de software, devido a inúmeras pontas soltas que foram abandonadas ou feitas as pressas.

Estimar um software é uma arte, imprecisa e cheia de armadilhas, que invariavelmente sacaneam com as equipes de projeto, não existe uma bala de prata para ajudar, porem as metodologias agiles tem ajudado a minimizar este problema.

Usem MS Project conheçam os caminhos críticos, atualizem em voo, reporte as anomalias e problemas que surgiram, o quanto antes identificarem o atraso, mais rápido se soluciona e criam alternativas.

Documentar os problemas

Evitar usar a estratégia de colocar um tapete por cima da sujeira, o ideal é documentar todas as falhas e problemas que surgiram ao longo do projeto, para educar a equipe e preparar o time para próximas adversidades.

Improvisar é bom, porem quanto mais situações anômalas, melhor enfrentaremos as próximas e o time não entraram em pânico ao primeiro sinal de fogo. Gerar uma base documental e criar treinamento em crises de software ajudaram a enfrentar o caos.

Cuidado na escolha dos comandantes.

Um projeto é o ponto alto de uma equipe, onde todos são cobrados até o limite, para isso devemos ter bons lideres, quanto mais importante for o projeto e maior visibilidade, melhor deverá ser o gerente, líder e Project owner.

Afinal o portfólio da empresa é feito por grandes clientes e grandes projetos, qualquer falha e a empresa está condenada, existe a lenda de uma velha e poderosa consultora que falhou catastroficamente em seus trabalhos, destruindo uma framework e afundando com alguns clientes.

Carlos Gomes dos projetos informáticos.

Pense que uma equipe de consultoria como os músicos de uma Orquestra, se cada um tocar seu instrumento ao bel prazer, teremos uma cacofonia de sons insuportáveis de ouvir, colocando um bom maestro, ordenando e colocando cada um na posição correta, seguindo a ordem correta, teremos sons celestiais.

A comunicação é importante, sabendo transmitir o conhecimento, passando a palavra, ouvindo e saber ser ouvido, o time sentira-se integrado e seguro, pronto para os desafios. Cada recurso deve ser alocado de acordo com sua capacidade e conhecimento.

Finalizando o Controle ideal.

Sem um mapa ou gps, não sabemos onde estamos, e não saber para onde vai não se chega a lugar algum. Um bom projeto deve-se a partida saber onde se quer chegar, quais os milestones e pontos de controle existem.

O líder deve acompanhar e atualizar os controles, evitando excesso de trabalho com controles impraticáveis e ao mesmo tempo se esmerando em controlar tudo dentro do razoável, auxiliando todos os elementos do time em prol do bem comum, ou seja, a conclusão do Projeto dentro do prazo, orçamento, escopo e qualidade.

Conclusão

Caro padawan o risco de software e a Crise de Software fazem parte de paradigmas avançados de programação, este modesto artigo teve como objetivo apresentar em linhas gerais vários pontos e auxiliar na trilha do conhecimento, fornecendo subsídios para pesquisas mais avançadas e compartilhar um pouco de minha jornada.

Concluindo sigam as boas práticas da linguagem escolhida, respeite os limites do Sistema, lembrando sempre que um sistema é parte de algo maior, recebendo inputs e fornecendo outputs para outros sistemas. Por isso respeite os limites e caso a complexidade aumente idealize e programe uma cisão de sistema.