

Porque empresas tratam as “Pessoas” como “Recursos”?

Adoro fazer perguntas, nem sempre consigo as respostas quando quero mais gradualmente vou conseguindo elas de varias formas, é só estar alerta. Eu acredito que é a qualidade das nossas perguntas que devemos nos concentrar para obter melhores respostas e aqui vou eu de novo. No artigo anterior “Você é uma Pessoa ou um Recurso” pelos comentários recebidos deu para ver que na maior parte das empresas de desenvolvimento de software as “Pessoas” são tratadas como “Recursos”; essa é a realidade, mais temos que entender o porque disto, as conseqüências negativas para o resultado dos projetos, pelo menos na nossa industria, as lições aprendidas e alternativas viáveis e mais eficazes.

Porque empresas tratam “Pessoas” como “Recursos”? Para entender o “porque” precisamos entender o contexto. Vivemos num momento de transformação da nossa sociedade de “Era Industrial” para a “Era da Informação”; muitos dos problemas de falta de emprego em outras áreas se deve a que a essência do trabalho mudou, é mais o menos o que aconteceu quando houve a mudança da “Era Agrícola” para a “Era Industrial” de uma hora para outra não era mais necessária tanta gente no campo para produzir o mesmo volume de produtos, porem faltavam pessoas qualificadas na industria. Agora com a automatização de grande parte do trabalho nas industrias esta sobrando gente lá, porem naquelas baseadas em conhecimento, como desenvolvimento de software, falta gente, falta muita gente altamente capacitada e experiente e existem ineficiências enormes. Na “Era Industrial” o foco central da economia era transformar materiais em produtos tangíveis através de processos definidos usando “Trabalhadores Manuais” que foram sendo gradualmente substituídos por maquinas que faziam o trabalho de transformação enquanto os trabalhadores acabaram assumindo o papel de supervisão da produção e auxiliando as maquinas a fazer seu trabalho. Na “Era da Informação” o foco é transformar dados em informações, as informações em conhecimento, e o conhecimento em sabedoria que agregue valor a indivíduos e/ou organizações de qualquer tipo. Computadores são ótimos para receber e processar dados, com bons softwares eles estão conseguindo transformar esses dados em informações, porem eles não conseguem transformar informações em conhecimento e muito menos conhecimento em sabedoria, ate onde eu sei os únicos capazes de fazer isso hoje são “Pessoas”. A produção de software é basicamente uma das industrias, que não produz absolutamente nada material, os “Recursos” necessários para permitir a transformação de insumos em produtos são intangíveis, basicamente “conhecimentos” e “habilidades” que estão dentro das “Pessoas” mais que para quem esta focado no processo produtivo elas não passam de “Recursos”. Então de aí vem que a gente vira “Recurso”, porque estamos integrados dentro do processo produtivo do software. Tratar as “Pessoas” como “Recursos” do ponto de vista do processo produtivo não seria problema, a menos que isso trouxesse conseqüências para os resultados. Quais são as conseqüências? Quando abstraímos e simplificamos o modelo e tratamos as “Pessoas” como “Recursos”; estamos tratando elas como coisas e elas são diferentes em vários aspectos por exemplo as “Pessoas” tem auto-determinação, coisas não tem, “Pessoas” tem varias partes de uma natureza corpo, mente, coração e espírito (consciência) com necessidades e motivações específicas, as coisas não tem. Para ilustrar, podemos tratar ou expressar qualquer coisa com uma maquina que ela não se importa, podemos delegar tarefas a maquinas sem que elas entendam qual o propósito ou contexto todo, ou pedir que façam coisas sem sentido, ou usar somente uma parte da sua capacidade, ou ser manipulativo e mau caráter e nada disto vai influenciar a sua capacidade de produção. Pessoas são diferentes, e no ignorar de essas diferenças é onde acabamos tendo todo tipo de conseqüências negativas para a produção eficaz, normalmente associadas com a capacidade de auto-determinação e variabilidade da eficácia das “Pessoas” que podem escolher de forma consciente ou inconsciente o nível de dedicação no trabalho [1] começando no menor grau que é a rebeldia ou desistência, passando pela obediência mal intencionada, o cumprimento voluntário, a cooperação animada, a dedicação profunda ou o mais alto grau de dedicação que é a empolgação criativa. Outro fator é que conhecimentos e habilidades poderão ser liberadas em maior ou menor grau dependendo das “atitudes” e neste quesito a gerencia tradicional de projetos fica refém do acaso já que não tem ferramentas para lidar com o assunto ou influenciar nele de forma eficaz, e a causa dessa falta de ferramentas em minha opinião esta no paradigma que esta sendo usando, a metáfora da “Maquina” e do “Recurso”, que simplesmente não permite que se entenda ou explore o problema em sua totalidade. Desenvolver software é basicamente uma atividade de transformação do conhecimento, poucas empresas, processos e culturas organizacionais estão adaptadas para essa realidade e tem a preocupação em criar o ambiente, infraestrutura e suporte efetivo para que equipes possam trabalhar de forma altamente eficaz. Normalmente qualquer iniciativa neste sentido é olhada dentro do paradigma dominante nas empresas como paparicar ou coisa similar. Outro elemento alem da dedicação e a auto-determinação é que o processo cognitivo das “Pessoas”; as habilidades de pensar e aprender, são extremamente influenciadas por três fatores, segundo a taxonomia de Marzano: as crenças sobre a importância daquele conhecimento digamos “eu quero ou preciso saber isto?”; as crenças sobre a eficácia digamos “será que consigo aprender ou entender?”; e as emoções associadas ao conhecimento que tem um impacto enorme na motivação para pensar ou aprender, digamos “o que sinto em relação ao que tenho que aprender ou pensar?”. Estes três fatores são decisivos para que efetivamente a pessoa possa ser eficaz pensando ou aprendendo, e muito do que tem que ser feito num projeto de desenvolvimento de software é pensar ou aprender. Quando tratamos as “Pessoas” como “Recursos” estamos reduzindo elas a “Coisas”; e acabamos manifestando isto quando achamos que podemos pedir para a pessoa para realizar uma tarefa, sem efetivamente explicar porque isto é necessário, sem criar um contexto para ela, qual o propósito, comunicando para ela que nos confiamos na sua capacidade e acreditamos que ela pode fazer aquilo, sempre usando

de sinceridade, de outra forma será percebido como falso e terá uma influencia negativa, e fazemos que o ambiente emocional do projeto seja positivo e “entregar produtos de qualidade” seja a “nossa” missão e possamos nos sentir orgulhosos de poder fazer isso com excelência de forma constante. Quando não fazemos isto, estamos criando obstáculos para que as “Pessoas” efetivamente possam realizar o seu trabalho, e algumas que tenham estratégias internas para ultrapassar estas barreiras conseguiram, as que não irão ter uma produtividade muito baixa e irão criar problemas para o objetivo do projeto. Além de entender que é necessário tratar as “Pessoas” como “Pessoas”, é imperativo entender que para produzir software precisamos de “Pessoas” cooperando com outras para um mesmo fim, isto é formar “Equipes”. Não são pessoas isoladas que produzem software e sim “Equipes” complementares onde as fraquezas de um membro da equipe são compensadas pela capacidades de outros. Lições aprendidas e melhores praticas Para desenvolver software de forma altamente eficaz é necessário formar “Equipes” de “Trabalhadores do Conhecimento” que tenham os “Conhecimentos” e “Habilidades” individuais e coletivos alinhados com suas “Atitudes”, suportados por processos que não interfiram em sua capacidade de se organizar para fornecer um produto. Formar equipes exige “Liderança” e para minha surpresa não é um dos requisitos para ser gerente de projetos o que dificulta muito o trabalho deles. Algum tempo atrás para fazer bom software a máxima era contratar os melhores profissionais e depois disso “delargar” para a equipe a missão deixando ela solta para entregar um produto, o que produzia resultados totalmente descontrolados, às vezes dava certo mais na maioria delas era um desastre total. Este modelo obviamente não funcionou e a solução natural da “Era Industrial” para este problema foi definir processos determinados, planejamento determinístico, detalhar as atividades, especializar recursos, gerenciar o processo, e controlar e verificar o cumprimento das atividades. Assim se inicio uma corrida no sentido contrario ás evidencias de como trabalhadores do conhecimento deveriam trabalhar, moldado no paradigma da “Era Industrial”. O que esta acontecendo neste momento na industria do software é que já se tem bastante evidencia que o modelo de comando e controle, processos definidos, planejamento deterministico, foco no controle do trabalho e não tem oferecido a melhoria nos resultados que se esperava e tem criado outros problemas. Muito daquilo que temos aprendido sobre gerenciamento de projetos não só continua valido mais é fundamental e de grande valor para os projetos de software, porem existe a necessidade de introduzir o que agora conhecemos sobre a natureza do software, o trabalhador do conhecimento, sobre pessoas e equipes para poder ser mais eficazes. Alternativas viáveis Ha alguns anos, vários lideres da industria de desenvolvimento de software se reuniram por alguns dias numa estação de ski nos Estados Unidos para discutir e analisar quais eram os princípios que estavam por traz de projetos bem sucedidos. Eles conseguiram chegar a um conjunto de valores e princípios comuns a todas as metodologias que eles estavam utilizando e criaram um manifesto ágil batizando assim uma serie de metodologias e uma abordagem radicalmente diferente da tradicional. Esta abordagem incorpora muitas das lições aprendidas na pratica e surge de um paradigma diferente mais adaptado ao trabalho em projetos de desenvolvimento de software. O mais interessante é que os resultados estão começando a ser medidos é tem se mostrado reveladores, muitos de nos já intuíamos e tínhamos experiência no tipo de resultados porem agora existem dados que permitem suportar esta opinião. Uma pesquisa recente com desenvolvedores de software nos Estados Unidos [2], com uma amostra de 4232 respostas, mostrou que 49% dos desenvolvedores pesquisados tinham adotado uma metodologia ágil, 65% tinham adotado pelo menos uma pratica ágil. Quando questionados sobre como uma abordagem ágil tinha afetado a sua produtividade, 12% reportaram que sua produtividade tinha sido muito maior, 48% um pouco maior, 34% não ouve mudança significativa, em 5% um pouco menor e em 1% dos casos muito menor. Sobre o quesito qualidade, foi perguntado como uma abordagem ágil tinha afetado a qualidade dos sistemas implantados 19% deles responderam que foi muito maior, 47% um pouco maior, 31% não ouve mudança significativa, 2% um pouco menor e 1% muito menor. Quando foi explorado a dimensão dos custos 2% dos entrevistados relatou custos muito menores, 20% um pouco menores, 53% não ouve mudança significativa, 22% um pouco maiores, 3% muito maiores. Sobre a satisfação dos usuários de negocio, 17% respondeu que foi muito maior, 41% um pouco maior, 39% não ouve mudança, 2% um pouco menor, e 1% muito menor. Existe uma correlação entre conhecimento e experiências negativas. Para entender 2.15% tiveram ao menos uma experiência realmente negativa, 16.74% tiveram alguma experiência relativamente negativa, o resto dos entrevistados não tiveram experiências negativas e muitos deles relatam projetos com melhores custos, mais produtivos, e produtos com maior qualidade. Conclusão Parece ser que a industria de software esta começando a encontrar soluções para o problema de 71% dos projetos terminarem como fracassos, estes números tem melhorado muito pouco desde o primeiro Chaos Report e talvez pela primeira vez na historia se estejamos encontrando respostas para as nossas perguntas. Por isso na verdade o importante não são as respostas, já que as respostas só podem estar ha altura das perguntas que fazemos, o importante melhorar na arte de fazer perguntas. Para quem quiser saber mais sobre metodologias ágeis existe um website da comunidade ágil no Brasil e pode ser encontrado aqui www.agilebrasil.com.br.

Sobre o Autor

Juan Esteban Bernabó é Fundador e Primary Consultant da TeamWare do Brasil uma empresa de consultoria focada em auxiliar organizações de desenvolvimento de software a otimizar seus processos, criar ativos de software, implantar processos ageis, utilizar de forma eficaz a tecnologias de objetos. Ele trabalha a mais de 15 anos em desenvolvimento de equipes de software altamente eficazes, definição de produtos, projetos, gerenciamento e coordenação de equipes de desenvolvimento, arquitetura de produtos, arquitetura de software, definição de politicas de reuso, avaliação de produtos

OpenSource para integrar dentro de software comerciais.

[1] 8o Habito “Da eficacia a Grandeza” – Dr. Stephen Covey [2] Survey Says: Agile Works In Practice – Scott Ambler