



«Fábrica de software» e saber tácito dos profissionais de informática

Leandra Leal¹, Francisco de Paula Antunes Lima²

^{1e2}Laboratório de Ergonomia – Departamento de Engenharia de Produção
Universidade federal de Minas gerais (UFMG)
CEP 31270-010, Belo Horizonte, MG, Brasil

{leal.leandra@ig.com.br, fpalima@ufmg.com.br}

Abstract. *The present work dissertates about the issues derived from the use of an organizational structure named software factory. Software manufacturers adopt such structure as a way to increase productivity and improve quality. The software development industrialization diverge from the primary creative activity in software production.*

Resumo. *O presente artigo discorre sobre os problemas decorrentes da utilização de uma estrutura organizacional denominada "fábrica de software", utilizada pelos desenvolvedores de software, na tentativa de aumento da produtividade e melhoria da qualidade de seus produtos, explicando que a industrialização do desenvolvimento de software contrapõe-se à natureza da atividade eminentemente criativa da produção dos software.*

1. Introdução

A “fábrica de software” é um modelo organizacional que começa a ser utilizado pelos fabricantes de software. O presente estudo aponta os equívocos desse modelo, que se consubstanciam na concepção de que o desenvolvimento de software seja “artesanal” e que a organização formal (regras, padrões, divisão do trabalho, hierarquia etc.) seja responsável pela “eficiência” da produção industrial.

Tentaremos demonstrar como a visão tecnicista desse modelo “amputa” a dimensão imaginativa e social do profissional em informática, desconsiderando a relação de dependência e interação entre os atores envolvidos no desenvolvimento de software, no que diz respeito ao saber tácito que envolve o processo criativo.

O estudo está sendo realizado em um setor de desenvolvimento de software - SDS de uma empresa prestadora de serviços públicos - SERVPUB. A pesquisa foi realizada juntamente com os profissionais de uma equipe que trabalha com um software que tem como objetivo atender, de forma integrada, uma rede de unidades com funções distintas: comerciais, operacionais e serviços de atendimentos aos clientes da empresa. Estes profissionais trabalham há doze anos com este sistema.

Este setor possui uma perspectiva de transformar-se em uma fábrica de software e de contratar, concomitante, uma empresa terceirizada, da mesma natureza, para a execução de alguns projetos de desenvolvimento de software parciais ou até mesmo integrais



O modelo "fábrica de software", que não é exclusivo da empresa sob estudo, segue a tendência atual da engenharia de software em utilizar modelos da organização industrial.

A metodologia empregada foi a da Análise Ergonômica do Trabalho (AET), fundamentada no ponto de vista da atividade, isto é, na atividade real de trabalho, que consiste em evidenciar o "trabalho real" em contraste com a organização formal (DANIELLOU et al., 1989). Foram observadas situações cotidianas no próprio local de trabalho, o que permitiu avaliar as reais condições em determinada situação de trabalho.

O nomes da empresa pública, do setor e da empresa terceirizada utilizados no artigo, são fictícios.

2. A empresa analisada

Fortalecer a empresa, consolidando a imagem de excelência na sua área de atuação é o principal objetivo da empresa estudada. Para que isto ocorra a sua grande meta, descrita em seu planejamento estratégico, é promover um "choque de gestão" seguindo o objetivo prioritário do Governo do Estado. Um dos pontos chaves dessa meta é a redução da relação despesas/receita do Governo Estadual.

Considerando-se a tendência das grandes organizações e a meta do governo, atualmente, conforme sugestão da gerência, a SERVPUB vem utilizando uma fábrica de software terceirizada - FABRISOFT para atender, em caráter parcial ou total, algumas de suas demandas e, concomitantemente, está implantando o conceito de fábrica de software para a equipe do SDS. Isto porque o setor é considerado como custo sem retorno em forma de capital para a empresa, além de apresentar vários problemas, tais como: baixa qualidade, baixa produtividade, incompreensão das necessidades do usuário, erros de construção, testes insuficientes e prazos quase sempre não cumpridos.

Na realidade esses problemas não são recentes e nem exclusivos da SERVPUB, tendo sido observado ao longo dos anos por vários autores que propugnaram soluções diversas para resolução desses problemas, [ver Carvalho,1988], sendo o modelo da "fábrica de software" apenas mais uma das tentativas utilizadas.

3. "Fábrica de Software": modelo inadequado para a solução de um problema real

O fundamento principal para a adoção deste modelo produtivo escuda-se na assertiva de que o trabalho de desenvolvimento de software é considerado "artesanal" e, por conseguinte, suscetível à industrialização. Os fabricantes de software buscam superar a construção de software "artesanal" reproduzindo os modelos industriais (Ver Greenfielta, Quidgest, Squadra, Meta, 2007)

"Fábrica de software utiliza processos formais baseados em conceitos industriais de planejamento, controle e produção. Por isso difere dos antigos bureais (*sic*) de programação, que constroem software de maneira artesanal". (Squadra, 2007).

O modelo organizacional da "fábrica de software" contrapõe-se ao da própria indústria, hoje a mercadoria padrão da era Ford não gera mais lucros extraordinários



para as indústrias, estas para sobreviverem nesse novo mercado têm que estar sempre inovando os seus produtos. Tal situação inviabiliza a separação entre a concepção e a execução, o que faz emergir um novo modelo organizacional em face da superação da organização taylorista/fordista.

Pode-se dizer de antemão que a fábrica de software é mais uma das tentativas de soluções ineficientes, pensadas a partir de um diagnóstico equivocado sobre a atividade dos desenvolvedores de software.

As fábricas de software são sempre inspiradas em modelos fabris de gestão de projetos, em certificação de garantia de qualidade do processo tais como CMMI (Capability Maturity Model Integration) e ISO (International Organization for Standardization) e na organização do processo de desenvolvimento segundo modelos industriais, a saber, RUP (Rational Unified Process) e XP (Extreme Programming). “O conceito de fábrica de software está baseado na idéia de prover uma linha de produção que atenda às necessidades específicas de cada cliente através da formalização de todas as atividades e seus produtos, trabalhando em linha de produção, com etapas e tarefas bem definidas para cada tipo de profissional, indo da produtividade da linha de produção a rotinas de controle de qualidade”. [BRITO, 2004].

Fernandes (2004) apresenta fábricas de software como “um processo estruturado, controlado e melhorado de forma contínua, considerando abordagens de engenharia industrial, orientado para o atendimento a múltiplas demandas de natureza e escopo distintas, visando à geração de produtos de software, conforme os requerimentos documentados dos usuários e/ou clientes, da forma mais produtiva e econômica possível”.

Em uma fábrica de software as pessoas estão agrupadas em estruturas funcionais, sendo que cada uma delas é responsável por uma parcela de trabalho ao longo do ciclo de desenvolvimento do serviço, conforme exemplificado no quadro abaixo.

Tabela 1: Divisões de funções na Fábrica de Software		
Setor	Unidades	Funções
Gerência		Gerente
		Secretária
PMO – Project Manager Office	PMO	Gerente PMO
		Assessor PMO
Setor de desenv.	Gerente Setor de Desenvolvimento	Gerente Projeto – Desenvolvimento
	Unid. Requisitos	Analista Negócio/Analista Sistemas
	Unid. Protótipo	Prototipador
	Unid. Projeto (desenho)	Projetista
	Unid. Implementação	Desenvolvedores
Setor de Manutenção	Gerente Setor de Manutenção	Gerente Projeto - Manutenção
	Unid. Requisitos	Analista Negócio / Analista Sistemas
	Unid. Projeto (desenho)	Projetistas



Setor de Apoio a Sistemas	Unid. Componentes	Adm. De Componentes
	Unid. Dados	Adm. De Dados
	Unid. Suporte ao Desenvolv.	Consultor Desenvolvimento
Setor de Controle da qualidade	Setor de Controle da Qualidade	Coordenador da Qualidade
	Unid. Adm. Qualidade Produto	Analista Processos
		Analista Controle Qualidade
		Coord. Célula de Teste
		Analista Testes e Usabilidade
	Testador	
Unid. Adm. Métrica/Indicador	Analista d PF	

Fonte: documentos interno da empresa estudada

Desde a década de 70 já se verificava uma tentativa de implementação da divisão do trabalho entre a concepção (tarefa de especificação) e a execução (tarefa de programação) inspirada no processo manufatureiro, que restou frustrada na sua aplicação. TAVARES já escrevia, desde 1983, que “a distinção feita pelas empresas entre analistas e programadores prende-se mais aos aspectos formais que a uma diferenciação funcional. As especificações das funções não são claras, mesmo quando algumas subcategorias são criadas (analistas de sistemas, analista de suporte, analista de O&M, programador sênior, júnior e trainee) as especificações ora se sobrepõem ou se confundem.”.

4. A atividade “artesanal” do desenvolvedor de software

O trabalho de desenvolvimento de software é considerado “artesanal” pelo fato dos produtos inteiros serem criados desde o início por um indivíduo ou pequenas equipes, “*sem documentar precisamente etapa por etapa todas as idéias e soluções parciais que vão surgindo ao longo do trabalho de elaboração de um sistema de informações*” (Carvalho, 1988:149), tornando esse(s) trabalhador(es) o(s) único(s) conhecedor(es) dos processos da construção do software.

No entanto, a atividade do analista não guarda qualquer similaridade com o trabalho artesanal, comparação possível apenas sob a ótica daqueles que demonstram um profundo desconhecimento da época pré-industrial.

Para se estabelecer uma diferenciação consistente entre o trabalho artesanal e o trabalho de desenvolvimento de software é importante conhecer melhor as características do trabalho artesanal.

O artesanato foi a forma de produção característica da Idade Média, na qual o artesão detinha os meios de produção (era o proprietário da oficina e das ferramentas) e trabalhando com a família em sua própria casa, “conduzindo todas as fases de produção de um objeto, desde a concepção até sua execução final”. [VARGAS, 1979]

O trabalho artesanal propriamente dito segue rigorosamente os padrões fundados nos costumes, que são passados de geração a geração, seguindo as tradições. As gerações mais novas recebem das mais velhas suas técnicas e todo o arcabouço de experiências acumuladas com uma ocupação denominada ofício. “Os trabalhadores



numa oficina estavam organizados de forma hierárquica, indo do aprendiz ao mestre, em vários níveis, até atingir o chefe. Todos viviam na mesma casa, participavam da vida do patrão até adquirirem também o grau de mestre, através de um trabalho considerado de alta qualidade” [CANÊDO, 1987].

Mas para garantir o segredo de seu ofício, o artesão da idade média não formalizava o processo produtivo, para dominar o saber e as regras de seu ofício, encontra-se atrelado a uma estreita convivência com os mestres durante vários anos. No início verifica-se a existência de uma divisão do trabalho, mas ao final do processo de aprendizagem, eliminam-se as diferenças e a hierarquia.

Esse ambiente possibilitou ao artesão um aumento ainda maior sobre todo o domínio do processo de trabalho, mesmo antes do início da produção, pois conta com a existência de uma forma de realizar o seu ofício, transmitida durante um longo processo de aprendizagem.

Como o trabalho artesanal segue uma tradição, o artesão não detém uma marca individual. Segundo Alexander (1969), o artesão figura como um agente da produção apenas reproduzindo o que lhe foi transmitido pela tradição. Não é necessária a introdução de inovações sem razão alguma, o que se permite fazer é o melhor que se pode dentro dos padrões preestabelecidos. As pequenas mudanças somente ocorrem em virtude de alguns ajustes necessários para sanar os defeitos que impõem correção.

De todas as características acima delineadas referentes ao trabalho artesanal, uma foi determinante para a origem do processo de industrialização, qual seja, a rigidez no processo produtivo, que facilitou a divisão bem delimitada das tarefas e a separação entre quem as planeja e quem as executa, traço característico da manufatura, precursora da atividade industrial, onde “a separação das tarefas era bem delimitada. Cada artesão era especialista numa ou noutra destas operações. Sua multiplicidade tinha a base da divisão do trabalho”. [BARRET, 1948: 58].

Na produção de software, diferentemente da artesanal, o analista não sabe aonde vai chegar. O profissional adquire o domínio do seu produto com o término de sua criação, não dispondo de padrões preestabelecidos, uma vez que cada software é desenvolvido em um processo produtivo diferente, único e específico que surge para atender as diversas demandas apresentadas no decorrer de sua produção. Por isso, o software se assemelha mais a um obra de arte [LÉVI, 1992] do que a um produto industrial.

O ritmo acelerado das mudanças técnicas é um fator preponderante para impedir que uma tradição se consolide, o que diferencia ainda mais a atividade do analista e deixa no produto uma marca pessoal, autoral.

5. A atividade de desenvolvimento de software

A criatividade dos desenvolvedores de software consiste em conseguir construir regras formais a partir das demandas específicas do usuário, considerando as situações singulares em que este atua. A máquina só é eficiente, quando ela interage bem com o homem, para isso é necessário que o profissional em informática identifique e considere as situações reais do trabalho humano. Assim, a geração de um software não é apenas a codificação de um conjunto de dados e regras, trata-se de uma atividade que exige transformar uma linguagem prática em uma linguagem formal, de códigos, o que torna



seu desenvolvimento complexo e difícil. Para Lévi (1992), uma das grandes dificuldades do programador, que gera um sofrimento gigantesco, é a de respeitar a sintaxe da linguagem da programação e produzir um “texto” coerente.

Diante da complexidade que envolve o desenvolvimento de software, tornam-se de difícil previsão todas as situações compreendidas em sua concepção. Diferentemente, a psicologia cognitiva entende que a concepção de um software é uma “resolução de problemas”. A resolução de problemas, sob a sua ótica, é tida como um processo de exploração ativa do “espaço do problema” constituídos de etapas (situação inicial, sub-objetivos e objetivo final). Assim, para se resolver efetivamente um problema considera-se necessário determinar uma seqüência de operações que transformam a etapa inicial em objetivo final.

Considerar a concepção de um software apenas como uma “resolução de problemas” é uma visão um pouco simplista. “Estudando em detalhe a maneira pela qual um profissional de informática resolve o «problema» que é a concepção de um software, nós constatamos que as coisas não se passam exatamente como pretende a teoria. Primeiramente, não somente o objetivo a atingir não é perfeitamente definido no início, mais ele sofre ao longo do desenvolvimento vários remanejamentos essenciais. Em seguida, os objetivos intermediários que levam ao objetivo final podem adquirir, ao cargo dos processos de resolução de problemas, um peso tal que todo os termos do problema são transformados.” [LÉVI, 1992, p. 44-5].

Os ajustamentos do software, durante e após o seu desenvolvimento, são vários. O processo de construção de um software exige do desenvolvedor, além do seu conhecimento técnico acerca da demanda, um manejo para conseguir articular o processo formal, traduzido em um programa de computador, à necessidade real do usuário, que é informal. O exemplo do vale serviço abaixo ilustra com clareza tanto o entendimento acerca das necessidades reais do usuário quanto os mencionados ajustamentos:

“...Aí, isso tudo é levantamento. Foram várias reuniões. Fui lá (no usuário) a gente conversou a primeira parte, aí quando eu comecei a fazer eu vi, opa! Tá faltando isso: aglomerado. Eu cheguei num ponto que eu fiz isso aqui... tava pra a empresa toda, só que aí tem um detalhezinho, eu falei pra ele (usuário) assim: é pra empresa toda? Não, é só pra aglomerado (usuário). Aí eu perguntei pra ele: o que que é aglomerado? E ele já tinha esse conceito lá, mas a gente não tinha informatizado o que era aglomerado. “...Como é que funciona? Aí eu fui desenvolvendo. Aí a gente desceu, conversou, oh! Participa de uma localidade, sei lá. Aí depois a gente vai bolando as telinhas. Aqui. Isso aqui é um cadastramento de um aglomerado. Aqui é pesquisa. Lá na agência de atendimento da Barroca tem esses aglomerados lá. Dentro desses aglomerados aqui eu tenho vários clientes lá dentro. Isso aí que a gente bolou. Como é que você quer enxergar isso aqui na tela? Ah, você me dá um nome e tal aí... Nós é que bolamos essa saída aqui oh. Primeiros ele informa a localidade. Vem pesquisando...porque? Porque primeiro eu tenho que ter a localidade, depois eu tenho uma agência de atendimento e depois eu tenho os aglomerados. Porque tudo tem uma seqüência. E os imóveis que fazem parte daquela localidade. O município é uma localidade. Hoje eu falo isso tudo, mas a gente reuniu várias vezes. A gente voltou lá e falou assim: desse jeito não vai dar certo. Vamos pensar pra este lado...” (Analista da SERVPUB)



6. O modelo fábrica de software e o saber dos profissionais de informática

O saber que será discutido aqui é o saber utilizado na ação, ele é contextualizado e situado. Na verdade existem algumas categorias de saber que são utilizadas na literatura: saber-fazer, saber-prático, saber que e saber tácito. Saber-fazer ou saber-como referem-se ao fazer do especialista, à sua habilidade e destreza em fazer determinada ação (Dreyfus e Dreyfus, 1989). Saber-prático são os saberes (teóricos e práticos) utilizados na ação (Vermersch, 1994), corresponde à regra aplicada na ação, é contextualizado e situado. Segundo Collins, (2001) saber-tácito “é o conhecimento que não é fácil de ser explicado, podendo ser uma habilidade motora, cognitiva, ou um método tradicional que não se sabe explicar porque funcionou”. O indivíduo não sabe explicar aquilo que sabe, ou não sabe como explicar, ou não sabe que sabe aquilo (FERREIRA, 2004).

Percebe-se nos profissionais de informática, com mais tempo e experiência, da empresa estudada, uma habilidade em desenvolver os programas dentro da expectativa do usuário, isso se deve ao saber prático e toda uma dimensão imaginativa e social desses profissionais, no que diz respeito ao saber tácito que envolve o processo criativo.

A nova forma organizacional (implantação do modelo de fábrica de software e terceirização), implantada na empresa, tem se constituído em um obstáculo para os técnicos em informática compartilharem os seus saberes. Tanto os empregados da SERVPUB quanto os empregados da FABRISOFT estão apresentando dificuldades em realizar as atividades dentro dessa nova estrutura organizacional. A separação dos membros da equipe em salas distintas e a distância da FABRISOFT, situada fora do estabelecimento físico da SERVPUB, são as principais queixas dos trabalhadores. Essas dificuldades se revelam pela insistência da analista de sistema da FABRISOFT em permanecer trabalhando no setor de desenvolvimento de software da SERVPUB, mesmo após o tempo de aprendizado previsto pelas empresas (SERVPUB e a FABRISOFT), que já se estende por mais de um ano após sucessivos adiamentos.

Como visto, o trabalho do analista envolve várias pessoas (usuários leigos e profissionais de informática) implicando interações de diversas naturezas e em diversos momentos. Como já falamos, o saber desse profissional se desenvolve em contexto particular, é situado e, para repassá-lo, é preciso que as pessoas compartilhem o mesmo ambiente, que tenham vivenciado a mesma situação, tornando a interação presencial entre os envolvidos imprescindível no momento de desenvolver soluções ou resolver problemas. Além do compartilhamento dos saberes, outros fatores exigem a interação entre os profissionais: dificuldade de memorizar um grande volume de informações; dificuldade de interpretar os documentos e a presença de documentos desatualizados. Os depoimentos abaixo demonstram essas situações:

Analista de sistema da FABRISOFT: “já imaginou quando eu sair daqui? eu preciso das pessoas o tempo todo para discutir, eu não conheço nada aqui. Eu sou direcionada por eles (empregados do setor de desenvolvimento), são eles que sabem a regra do negócio. Eu não consigo estar fora daqui, o conhecimento está aqui, o ambiente está aqui, o apoio está aqui, só o desenvolvimento está lá”.

Analista de sistema da FABRISOFT: “O pessoal da equipe não deixa eu ir para lá, dizem que eu produzo mais aqui. Aqui já está difícil eu achar as pessoas, já imaginou lá na fábrica. Eu estou precisando de falar com fulano mas não consigo”



Analista da SERVPUB: “estão passando para ela (analista terceirizada) uma demanda de faturamento, mas ela não sabe nada. Se ela ficar aqui a gente pode discutir e eu vou passando aos poucos as informações. Imagina ela longe anotando em um papel, como foi falado na palestra, e tendo que vir aqui trazer o papelzinho, quando ela voltar já vai estar cheia de dúvidas de novo. Olha eu já conheço a muito tempo o faturamento e ainda tenho dúvidas”

Analista da SERVPUB: “Dizem que não precisa da gente passar as informações porque está tudo registrado, mas o sistema é muito grande, isto não é possível. Mesmo com toda a documentação, a pessoa tem dúvida e vêm aqui”.

Analista de sistema da FABRISOFT “Para eu conhecer um sistema de faturamento tenho que ficar pelo menos um ano, isto aqui é muita informação”.

Além das dificuldades impostas pela distância física da empresa FABRISOFT, a nova estrutura organizacional apoiada em uma divisão de tarefas, distribuídas e sequenciais tal como em uma linha de produção, também é um fator preponderante que distancia ainda mais os técnicos e o usuário, impedindo que os saberes sejam compartilhados. Ademais, conforme se afere nos depoimentos descritos, estas mudanças organizacionais estão dificultando a realização das atividades, principalmente quando se depara com situações imprevisíveis. Um exemplo marcante desta dificuldade é a demanda do serviço (desenvolvimento e manutenção) que só pode ocorrer através de uma única via, abrindo-se uma solicitação de serviço pela intranet.

“A gente não recebia ordem de serviço. Se a gente espera a ordem de serviço, às vezes a gente não vê o problema”. (Analista da SERVPUB)

“É muito melhor a gente atender o cliente na hora, por telefone, a gente pega o problema no momento em que ele está acontecendo. Um exemplo é a função do atendimento. Essa função era utilizada 1500 vezes por dia e nesse intervalo ocorria uma pane que parava tudo. Na hora que acontecia esse problema me avisavam e com isso consegui identificar o terminal e o atendente que trabalhava com ele. Descobri que esse atendente era novato e ele utilizava um campo pulando outro que era obrigatório preencher antes deste. Ele encontrou uma forma para fazer isto. A fábrica terceirizada não iria descobrir nunca e nem a gente se não fosse na hora”. (Analista da SERVPUB)

7. A construção do saber dos analistas e programadores da empresa estudada

Os analistas e os programadores da equipe do sistema integrado estão aproximadamente há doze anos realizando manutenções e novos desenvolvimentos para o software integrado (comerciais, operacionais e serviços de atendimentos aos clientes da empresa). Nesse meio tempo foi possível criar um conhecimento entre os programadores e analistas de tal forma que foi possível desfazer-se essa divisão entre os profissionais. Esta equipe da SERVPUB, como ocorre em outros desenvolvimentos de software no setor, apresentou vários problemas (demandas atrasadas, erros de construção e incompreensão das necessidades do usuário) no decorrer desses anos, como demonstram os depoimentos abaixo:

“Temos demandas com mais de um ano atraso” (Analista da SERVPUB)



“Tem usuário que parece que quer que dê errado, nunca informam tudo” (Analista da SERVPUB).

A representação do analista em relação usuário é equivocada. O que de fato acontece não é má vontade ou boicote dos usuários, mas sim dificuldades inerentes ao diálogo entre um projetista especializado e um usuário leigo, que não pode, por definição, verbalizar tudo o que precisa e precisará em um produto a ser desenvolvido (para mais detalhes, ver Ferreira, 2004) Apesar desses problemas, o software tem conseguido atender as necessidades do usuário, isto porque, a forma organizacional, separando a concepção da execução, não foi mantido pelos profissionais dessa equipe, o que facilitou a interação entre estes trabalhadores e a construção do conhecimento coletivo. O depoimento abaixo ilustra essa situação:

“Não existia a diferença entre analista e programador, fomos construindo um conhecimento único, não tem jeito de separar. É muito difícil, o computador não tem meio termo, se alguém mandou ele fazer errado, ele vai fazer errado. A comunicação é muito complicada, se a gente entende errado a gente vai fazer errado”. (Analista da SERVPUB)

Os profissionais de informática da SERVPUB tinham um acesso fácil aos usuários. Com esta facilidade de interação e com tempo que trabalhadores estão envolvidos nos setores comercial e operacional, cada indivíduo dominando mais uma área específica, foi conhecendo a necessidade real dos usuários.

Com o conhecimento adquirido por estes trabalhadores, o software foi sendo aprimorado e, a cada dia que passa, está sendo considerado mais eficiente e com melhor qualidade. Esta eficiência só não é melhor devido às pressões temporais vivenciadas pelos técnicos, que dificultam que as pessoas compartilhem os seus saberes, e devido ao desconhecimento e/ou desprezo da empresa do papel fundamental do saber coletivo no processo de desenvolvimento de produtos, sejam materiais ou imateriais.

Mesmo com a atual distância geográfica existente entre os técnicos, eles sempre encontram formas alternativas de se comunicar, como assegurado pela permanência da analista da empresa terceirizada na empresa pública, os profissionais da equipe se deslocam para as salas de seus colegas para discutirem determinado caso, etc. É através destas interações que os profissionais constroem e dividem os seus conhecimentos. Vários são os diálogos que ocorrem entre os profissionais da equipe e o usuário no decorrer de um desenvolvimento ou manutenção do software, como ilustrado por parte de um diálogo entre o analista da SERVPUB e a analista da FABRISOFT durante a execução de uma atividade de teste de um programa que realizará um relatório do esgoto de Montes Claro.

Programadora da FABRISOFT: “Eu pedi o grupo 101 e não apareceu nada. Eu não sei se as matrículas estão corretas? Ele não estava listando ninguém”

Analista da SERVPUB: “Ele não ia listar mesmo, está certo porque a referência é nove”.

Programadora da FABRISOFT: “Vamos ver na referência 8 se está certo”

Programadora da FABRISOFT: “Olha aqui também não tem”



Analista da SERVPUB: “Olha lá este faturamento é avulso e por isto não tem.”

Programadora da FABRISOFT.: “Então vamos ver outro...é este deu certo”

Programadora da FABRISOFT: “É isto mesmo que você queria?”

No início, a programadora não percebeu que estava utilizando a referência errada, tendo o analista identificado prontamente esse erro. Logo em seguida, a programadora solicitou aleatoriamente uma matrícula na referência 8, entretanto tratava-se de um faturamento avulso e por isto o teste não dava certo. A programadora não conseguiu identificar na tela que o faturamento era avulso, e o analista somente reconheceu esta situação quando viu na tela que a fatura não apresentava água nem esgoto, o que a diferenciava da fatura normal.

“Quando a gente não acha o que procura, a gente bate o olho e vê as características na tela e percebemos do que se trata” (Analista da SERVPUB)

Dessa forma, através da interação presencial, a competência e a experiência vão sendo compartilhadas para que os trabalhadores consigam desenvolver software que atendam cada vez mais às necessidades do usuário.

Para se desenvolver um software integrado que atenda aos usuários, os técnicos têm que possuir o conhecimento real das necessidades daqueles. Os próprios trabalhadores percebem esta necessidade, que aqui se encontram evidenciadas através de um depoimento:

Analista da FABRISOFT: “...é muita informação, para eu conhecer um sistema de faturamento tenho que ficar pelo menos um ano”.

Os benefícios da conjugação das formas de saber podem ser confirmados na empresa através das atividades desenvolvidas por seus técnicos, que possuem doze anos de experiência adquirida através da interação entre os técnicos e os usuários, quer para compreender as demandas desses usuários, quer para dar-lhes orientações de como proceder. O técnico consegue, com o auxílio de sua experiência anterior, atender a demanda de manutenção ou de um novo desenvolvimento com maior rapidez e eficiência. Isto porque estes trabalhadores já têm um conhecimento das necessidades dos seus usuários. O seguinte exemplo ilustra bem este conhecimento:

Foi solicitado à FABRISOFT, pela SERVPUB, o desenvolvimento de um relatório das atividades realizadas pelos seus leitores através do aparelho pocket. Após o início dos trabalhos pela terceirizada, foi repassado à SERVPUB um outro preço para dar continuidade ao desenvolvimento do software. Como o preço foi considerado muito alto pela SERVPUB, esta resolveu repassar a tarefa para o seu setor de desenvolvimento.

A FABRISOFT repassou à SERVPUB todos os documentos pertinentes às etapas do trabalho que desenvolveu. Durante a avaliação das informações descritas nos documentos, o analista e o programador da SERVPUB, responsáveis pela demanda, realizaram várias críticas sobre ele:

Analista- “Eles pediram 3 meses para fazer o relatório, com certeza faríamos muito mais rápido, isto é lógico já conhecemos bem o sistema, iríamos queimar etapa”



Programador - “Vimos que eles estavam indo para um caminho que iria dar problema. Colocaram no relatório o início da rota descrita no aparelho sileim para os leiturista, mas já sabemos que isto não dá certo, porque nem sempre isto acontece, as vezes ele pode começar por exemplo pela rota 3, o certo seria colocar hora do início do trabalho e hora do fim do trabalho. O relatório não é por rota”.

Analista- “Estou vendo que a fábrica terceirizada fez uma coisa a mais. Eles criaram um processo só para o registro de ociosidade, eles estão dando uma volta a mais que não precisava. Como eu sempre digo tem vários caminhos para chegar ao centro, mas você pode ter que dar mais volta para chegar.

“Estou retirando este processo de ociosidade a mais que eles redefiniram para não precisar de fazer uma comunicação a mais entre o POCKET e o servidor. A nossa empresa paga por cada comunicação. Estou aproveitando a comunicação da própria leitura para mostrar a ociosidade. Na realidade eu mudei o nome não chamo de ociosidade, mudei para intervalo entre leitura. Eu achei pesado colocar ociosidade, sei que não iria ficar bom para os leituristas, já sei o que eles pensam. Então perguntei para o gerente da leitura e ele achou melhor colocar intervalo entre leitura. O registro de ociosidade iria mostrar apenas o tempo de intervalo sem executar a leitura e não iria ter a hora que iniciou e finalizou, e é claro que é isso que o gerente quer. Exemplo, ele fez 15 minutos de intervalo, mas qual foi a hora que ele iniciou e a que finalizou o intervalo”.

Neste caso, o analista e o programador, com o conhecimento real da atividade dos leituristas, puderam antecipar a necessidade do cliente, que precisaria saber o tempo de intervalo entre as leituras e o horário de início e término de cada leitura.

Analista “Estou tendo que fazer o serviço de novo. Assim que eles começassem a rodar o relatório iriam aparecer os problemas. Tive que mudar o leiaute. Eu conheço mais a empresa sei dos detalhes, já fiz outros relatórios”.

O desenvolvimento de software se desenrola através de interações subjetivas entre os agentes envolvidos, e por isto a atividade não pode ser toda formalizada, pois o conhecimento somente é adquirido através da socialização desses agentes dentro de um contexto compartilhado.

8. Conclusão

O modelo produtivo proposto pela “fábrica de software”, calcado na idéia equivocada de que a ES seja "artesanal" e que a eficiência da produção industrial deve-se inteiramente à organização formal (regras, padrões, divisão do trabalho, hierarquia etc.), não se coaduna com a natureza da atividade exercida pelos desenvolvedores de software, seja pelo elemento criativo inerente ao exercício dessa atividade, seja pelo traço característico da unicidade e exclusividade do processo produtivo empregado no desenvolvimento de cada software, ou ainda pela necessidade de interação, em todas as etapas do desenvolvimento, entre os analistas, programadores e usuários.

Dissociada, portanto, da idéia inicial do trabalho “artesanal”, a atividade do analista assume contornos de natureza artística onde os produtos desenvolvidos são únicos, a exigüidade de tempo e recursos é suplantada por soluções eminentemente criativas, as



escolhas pessoais não obedecem a tradições, as atividades desenvolvidas no processo produtivo são indissociáveis e incompatíveis com o modelo formal de divisão do trabalho, o que sugere uma indefinição nas normas de produção, de qualidade, de tempo, incompatível com o modelo de organização fabril.

9. Referência bibliográfica

- ALEXANDER, C. Ensayo sobre la syntesis de la forma. Buenos Aires: Ediciones Infinito, 1969.
- ANTONETTIG. A Economia Medieval. São Paulo: Atlas, 1977.
- BARRET, F. Histoire du Travail. Presses Universitaires de France, 1948.
- BRITO e BRITO. Metodologia para Gestão do Processo de Qualidade de Software para Incremento da Competitividade da Mobile. Disponível em: <<http://www.mct.gov.br>>. Acesso em 25/05/2006.
- GUÉRIN, F. et. al. Compreender o trabalho para transformá-lo. São Paulo: Edgard Blucher, 2001.
- CANÊDO, L.B. A Revolução Industrial. São Paulo: Atual: Campinas: Unicamp, 1987.
- FERNANDES e TEIXEIRA. Fábrica de software: implantação de gestão de operação. São Paulo: Atlas, 2004.
- FERREIRA, R.B. “Diálogos de surdos: a difícil explicitação do saber entre programadores de software e operadores de fábrica”. Dissertação de mestrado pela Engenharia de Produção: UFMG, 2004.
- FONSECA, G.C. Aprendizagem, socialização e conflito no trabalho: a dimensão tácita do (des)conhecimento nas organizações. Belo Horizonte, 2005. 113f. dissertação (mestrado em Engenharia de Produção) – Escola de Engenharia de Produção, Universidade Federal de Minas Gerais.
- GORZ, A. O imaterial: conhecimento, valor e capital. São Paulo: Annablume, 2005.
- LATOUR, B. e Woolgar, S. A vida de laboratório. Rio de Janeiro: Relume Dumará, 1997.
- LÉVI, P. De la programmation considérée comme une des beaux-arts, Paris, La Découverte, 1992.
- RUGIU, A.S. Nostalgia do mestre artesão. São Paulo: Autores associados, 1998.
- TAVARES, S.R.S. Da crise do software ao projeto estruturado: a submissão não real do trabalho em programação. In: FLEURY, A.C.C.; VARGAS, N. Organização do trabalho. São Paulo: Atlas, 1983.