

Um Olhar Sociotécnico sobre a Engenharia de Software

Henrique Luiz Cukierman¹

Cássio Teixeira¹

Rafael Prikladnicki²

Resumo: As novas tecnologias modificam a forma e a substância do controle, da participação e da coesão social. Porém, ao fazê-lo, são também modificadas pela experiência social, de sorte que o técnico e o social constituem um movimento de "co-modificação", somente percebido por uma aproximação concomitantemente social e técnica, por um olhar sociotécnico. O artigo pretende apresentar algumas das principais características deste olhar, bem como discutir os desafios que coloca para a engenharia de software.

Abstract: New technologies modify the form and the substance of social control, participation and cohesion. However, as they modify, they are also modified by social practices in such a way it is possible to argue that social and technical dimensions constitute a process of mutual construction, only apprehended through an approach simultaneously social and technical, through a sociotechnical frame. This article presents some of this frame's main features, as well as its challenges to software engineering.

Palavras-chave: Engenharia de Software, Olhar Sociotécnico, Interdisciplinaridade, WOSSES

¹ PESC/COPPE/UFRJ
{hcukier, cant}@cos.ufrj.br

² FACIN/PUCRS
{rafaelp@puers.br}

1 Introdução

É necessário, para não dizer urgente, ter em conta o novo ordenamento social que se está produzindo desde o surgimento e a adoção das novas tecnologias de informação e comunicação. Investigá-lo é crucial, seja para descortinar novas perspectivas de sucesso comercial e empresarial, seja para construir uma melhor qualidade de vida e uma sociedade mais justa. Essas novas tecnologias são reputadas como fontes de mudanças radicais, e neste caso, constituem um cenário no qual transformam significativamente várias dimensões da vida moderna, entre outras a natureza e a experiência das relações e comunicações interpessoais, as relações e condições de trabalho, o modo de funcionamento do mundo dos negócios, da cultura e das artes, ou ainda a formulação de políticas regulatórias adequadas. Em síntese, as novas tecnologias modificam a forma e a substância do controle, da participação e da coesão social. Porém, ao fazê-lo, são também modificadas pela experiência social. É o caso da ES (Engenharia de Software) e, de fato, sua comunidade acadêmica e profissional usualmente destaca a importância de questões sociais, culturais, políticas e organizacionais nos projetos de desenvolvimento de software e nos projetos de implantação e melhoria de processos de software. Segundo FUGGETTA [1], *"pesquisadores e praticantes têm percebido que desenvolvimento de software (...) é um esforço coletivo, complexo e criativo. Deste modo a qualidade do produto de software depende fortemente das pessoas, organizações e procedimentos utilizados para criá-los e disponibilizá-los"*.

No entanto, essas questões não recebem, em intensidade compatível com o reconhecimento de sua importância, a atenção devida, nem na literatura nem nos eventos sobre ES. É muito comum vê-las qualificadas como questões "não técnicas", fazendo supor que a maior parte da comunidade de ES acredita que é possível dividir os problemas em "técnicos" e "não técnicos". Todavia, vai ficando cada vez mais claro que tal divisão não dá conta de enfrentar os crescentes desafios da ES. SOMMERVILLE [2] sugere que *"gerenciamento efetivo [de projetos de software] trata [...] da gerência das pessoas na organização. Gerentes de projetos têm que resolver problemas técnicos e não técnicos através das pessoas alocadas em suas equipes da maneira mais efetiva possível"*. De imediato, essa divisão nos remete à idéia de que aqueles problemas relativos à primeira classe – os técnicos – dispõem de maior importância. À segunda classe – os não técnicos – não se confere sequer uma qualificação positiva. São definidas pela exclusão, pela negativa: não técnico.

Pouco qualificadas, e ocupando um espaço relativo muito menor, as questões sociais, culturais, políticas e organizacionais são relegadas a disciplinas externas à ES. LEVESON [3] observa que:

o fervor e a excitação na busca de se desenvolver uma tecnologia revolucionária, com potencial de mudar o mundo de forma profunda, pode estar concentrando [a discussão] apenas no técnico e excluindo o social. [...] Nos primeiros 50 anos vimos o desenvolvimento do conceito de software como um produto de engenharia e um objeto matemático, mas pouca atenção foi devotada ao software como um produto humano. [...] Talvez esse viés 'técnico' possa explicar a grande evolução da ES nas fases finais do ciclo

de vida de desenvolvimento de software. [...] Poucos cientistas da computação têm se preocupado em considerar o tremendo efeito da tecnologia sobre a vida e sociedade humanas; é fundamental para a ES, atentar para esta situação [...] reconhecendo a importância e estreitando laços com as ciências sociais e a psicologia cognitiva.

O desenvolvimento de sistemas de software parece envolto em uma "ortodoxia técnica", e, portanto, visto apenas como um processo "técnico", a ser realizado por especialistas. Todavia, o esforço necessário para o desenvolvimento desses sistemas apresenta problemas e desafios de complexidade muito além da técnica, ou seja, que exige a intervenção de saberes diferenciados, oriundos de outras áreas do conhecimento. Por essa razão, a comunidade de ES não pode deixar de se "contaminar" pelas contribuições das ciências humanas e sociais. Felizmente, muitos pesquisadores têm percebido que a tecnologia da informação também é inevitavelmente social e, com isso, têm tentado focalizar a ES como um problema concomitantemente de complexidade social. HERBSLEB [4, pp.23-24] expressa tal percepção de forma cristalina:

Uma compreensão mais profunda do nosso próprio campo nos conduz à interseção de várias disciplinas científicas – um espinheiro emaranhado intelectual que muitos dentre nós preferem evitar a destrinchar. Compreender como se pode melhorar a engenharia de software requer um aprofundamento da nossa compreensão quanto a duas dimensões: 1) a dos princípios e práticas efetivos em engenharia de software; 2) a de como tais princípios e práticas se alinham frente ao modo como seres humanos funcionam cognitiva, social e culturalmente. Em minha opinião, a pesquisa atualmente realizada em engenharia de software tem progredido de forma estável quando se trata da primeira dimensão, porém correndo o tempo todo o risco de resultar irrelevante ao negligenciar as realidades interpostas pela segunda dimensão. Tendemos a assumir que seres humanos poderão mudar, e simplesmente o farão de todas as maneiras que se mostrarem necessárias. Todavia, o funcionamento humano não é assim tão maleável, e, para imenso prejuízo de nossas pesquisas, ignoramos tal fato.

A saída para os impasses criados pela separação entre o técnico e o social consiste em mudar o ângulo de aproximação e, assim, percebê-los por um novo enquadramento. Resumindo brevemente, um enquadramento em que o técnico e o social constituem um movimento de "co-modificação", somente percebido por uma abordagem concomitantemente social e técnica, por um **olhar sociotécnico**. Um olhar que busca apreender a ES sem fragmentá-la em "fatores ou aspectos técnicos" de um lado, e "fatores ou aspectos não-técnicos" de outro, sem fatorá-la em quaisquer outras dualidades ("fatores técnicos" versus "fatores humanos, organizacionais, éticos, políticos, sociais, etc.") que terminem por desfigurar o "pano sem costura" que imbrica na ES o técnico e o social em um mesmo e indivisível tecido³.

³ Apesar de algumas similaridades, são distintas a abordagem sociotécnica proposta no presente artigo e o projeto (*design*) sociotécnico de sistemas oriundo da 'Tavistock School' do início dos anos 1950. Mais intensamente aplicada a TI a partir dos anos 1970, a abordagem da 'Tavistock School' propõe que um melhor projeto de sistemas de informação somente pode ser alcançada quando se leva em conta as necessidades sociais dos usuários.

A comunidade internacional da ES já vem dando mostras de reconhecer a importância de alcançar além dos assim chamados "aspectos técnicos". As citações até aqui apresentadas constituem uma evidência a respeito, embora nenhuma delas seja de autor (a) brasileiro (a). Em verdade, ainda não se tem conhecimento de uma abordagem especificamente brasileira a respeito do olhar sociotécnico direcionado à ES, de sorte que podemos reivindicar a condição de pioneiros. Não é por outra razão que já era mais que justificada a instituição em nosso país de um fórum ampliado para o debate onde fosse possível, ainda que de forma incipiente, explicitar e tratar uma vasta gama de questões que, por ultrapassar as fronteiras das tecnicidades, era normalmente relegada a um segundo plano, desconsiderando assim a complexidade sociotécnica do objeto da ES. Este fórum consubstanciou-se através da realização de três edições do WOSES – *Workshop Um olhar sociotécnico sobre a Engenharia de Software*¹, de cuja organização participamos, cujo intuito foi, e continua sendo, o de contribuir para a ES brasileira de tal forma que nossas ferramentas, métodos e processos recebam insumos que nos permitam produzir e manter, com maior eficiência e qualidade, nossos sistemas de software. Ao consubstanciar-se um novo olhar, o olhar sociotécnico, a pretensão é a de caminhar na direção da construção de uma ES que, dada a ubiquidade dos produtos de software, contribua efetivamente, para prover uma consciência crítica da produção de ciência e tecnologia frente à sociedade brasileira, e, portanto, da construção de um país mais justo e mais democrático. De forma geral, os objetivos do WOSES podem ser assim condensados:

1. Promover novas e melhores formas de interação entre o técnico e o social, buscando superar fronteiras entre a ES e outros saberes, especialmente aqueles oriundos das ciências humanas e sociais;
2. Buscar uma nova compreensão do sucesso/fracasso dos projetos de desenvolvimento, implantação e melhoria de processos de produção de software à luz das relações éticas, sociais, políticas, econômicas e históricas indissociáveis da prática da ES;
3. Encetar os primeiros passos para a formação de uma rede de pesquisadores brasileiros interessados pelo desafio de construir uma abordagem sociotécnica da ES, procurando socializar as experiências dos grupos de pesquisa brasileiros já envolvidos com o tema, bem como estimular a formação de novos grupos;

Todavia, trata-se de uma abordagem onde ainda predomina a dicotomia entre o técnico e o social que, em nossa proposta, inspirada a partir das correntes mais contemporâneas dos estudos interdisciplinares sobre a ciência e a tecnologia (referidos no mundo acadêmico de língua inglesa como *Science and Technology Studies*), procuramos superar. Para uma recente revisão crítica dos diferentes movimentos em torno das proposições sociotécnicas, veja, por exemplo, Brigham and Introna [27], e para uma contextualização histórica de diversas linhas de pesquisa sobre aspectos sociotécnicos da computação ver [11].

4. Entender a atual configuração da ES no Brasil, através da contextualização histórica de seu ensino e prática;
5. Contribuir para a produção de novos saberes capazes de enriquecer o debate sobre a ES, potencialmente aptos a agregar eficiência e qualidade ao desenvolvimento, manutenção e implantação de sistemas de software.
6. Construir e fortalecer os vínculos entre a abordagem sociotécnica e a demanda por qualidade de software, especialmente à luz de estudos de caso de desenvolvimento, manutenção e implantação de sistemas de software no Brasil.

O presente artigo pretende não somente colocar os seus “dois tostões” na formulação de um olhar sociotécnico que possa vir a contemplar as especificidades brasileiras⁴, mas também rever as principais linhas de discussão que animaram estas três edições do WOSES.

2 Os Desafios de um Olhar Sociotécnico

Imbricação, indissociabilidade e indeterminação do técnico e do social fundamentam o olhar sociotécnico, o qual, por não separá-los aprioristicamente, concebe-os, o técnico e o social, como uma mútua determinação, a exemplo do que é sugerido pela ilustração de M.C. Escher, reproduzida na figura 1.

⁴ Cabe esclarecer que se trata muito mais de intenção e desejo do que propriamente de uma proposição concreta a respeito do que seria um olhar sociotécnico brasileiro. Dito de outra forma, ao invés do estabelecimento de metas, o que está em jogo é a formulação de alguns marcos de partida, até porque se assim não fosse, seria um empreendimento demasiado pretensioso e de pouco fundamento, uma vez que há uma imensa carência de material especificamente brasileiro a respeito. Portanto, por tratar-se aqui de uma discussão de princípios e não propriamente de resultados, não é nossa proposta, apesar de constituir-se como uma promissora via de estudos, investigar semelhanças e diferenças da realidade brasileira frente à realidade de outros países.

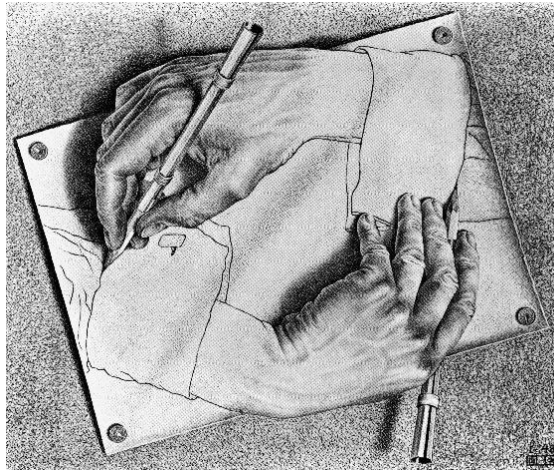


Figura 1 – Uma escrita mútua
(M.C. Escher's "Drawing Hands"(c) 2008 The M.C. Escher Company - the Netherlands. All rights reserved. Used by permission⁵. www.mcescher.com)

A imagem indica que há uma escrita mútua, pela qual uma das mãos, no mesmo movimento em que desenha a outra, por ela é igualmente desenhada. Desenhar e ser desenhado se confunde e se mescla, tornando impossível dissociar ambas as ações. Pode-se avançar nas metáforas inspiradas pela obra do artista, propondo que, assim como as mãos podem representar as relações e interações entre o técnico e o social, podem igualmente fazer pensar em uma escrita que é ao mesmo tempo a das ciências exatas e a das ciências humanas e sociais. Uma escrita interdisciplinar (ou mesmo transdisciplinar) por excelência.

Da escrita e determinação mútuas, emerge uma nova imagem, a da complexidade, seja a do conhecimento que se quer construir, seja a das ferramentas para construí-lo. Em meio a múltiplos traçados, e especialmente em meio à indeterminação de quem/o que constrói e quem/o que é construído, a questão que se coloca é a de como lidar com a complexidade na ES. Não por acaso este é o próprio título do trabalho apresentado no WOSES 2005 por AUDY & PRIKLADNICKI [5], cuja proposta repisa com muita propriedade os limites da disciplinaridade, apontando a via da multi/inter/trans disciplinaridade como a saída para se avançar na construção do conhecimento em ES. Ambos os autores retornaram ao WOSES em 2006 com a mesma temática, desta feita relacionada aos desafios práticos colocados pelo desenvolvimento global de software (caso particular do desenvolvimento distribuído de software – DDS), decorrentes da crescente transformação de mercados nacionais em mercados globais, tornando assim possível novas formas de cooperação e competição que vão além das fronteiras dos países. Nesse artigo, os dois professores da PUCRS evidenciam as dificuldades para se analisar o fenômeno do desenvolvimento global de software caso contassem única e exclusivamente com os métodos

⁵ M.C. Escher's Drawing Hands. Todos os direitos reservados. Utilização permitida.

e ferramentas da ciência da computação, como seria de se esperar a partir da própria localização do seu departamento naquela universidade, conforme mostra a figura a seguir [6]:



Figura 2. Localização do DDS na PUCRS, segundo o olhar “técnico”

Como o DDS é uma área que envolve o estudo interdisciplinar das teorias e práticas de diversas outras áreas, de modo a contribuir com a ainda pequena, mas crescente base de conhecimento sobre desenvolvimento distribuído de software, a figura abaixo ilustra como o DDS poderia ser visto, tendo por base não mais as tradicionais pertinências disciplinares, mas sim as próprias características e desafios de pesquisa.

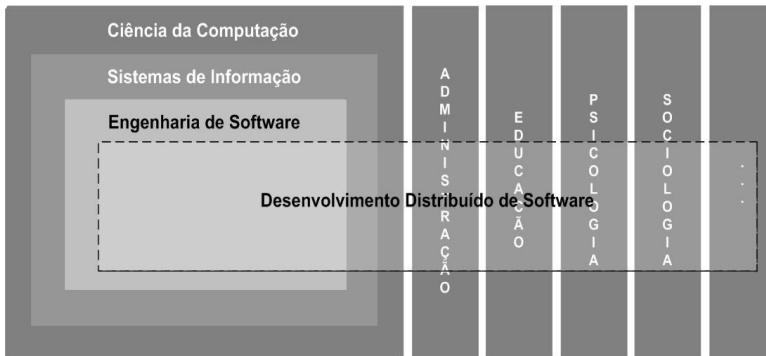


Figura 3. Localização do DDS na PUCRS, segundo o olhar sociotécnico

A partir da figura 3, entende-se que o olhar sociotécnico passa a existir no momento em que o DDS é entendido a partir da visão de não apenas uma, mas diversas disciplinas ou áreas do conhecimento que se complementam, indo além da simples integração entre as disciplinas. Desta forma, o que existe não é apenas a integração, mas também um diálogo comum entre as diversas áreas, tornando o DDS a atividade que integra diferentes visões de engenheiros de softwares, administradores, educadores, psicólogos, sociólogos, entre outros.

Certamente que a empreitada de fazer dialogar saberes tão distintos pode mostrar-se extremamente dificultosa, não sendo, todavia impossível, e, portanto, é recomendável que seja experimentada. Por exemplo, MENDONÇA [7], em seu artigo *Usabilidade de Processos*, abordando uma questão mais pontual e localizada, exercita o diálogo entre ES e ergonomia quando discute a aplicação do conceito de *usabilidade* na prescrição dos processos de software. Além disso, um exemplo oportuno foi apresentado no WOSSES 2007, relacionado com o dia a dia das atividades de desenvolvimento de software e a necessidade de um viés sociotécnico. No artigo *Teoria da Atividade: Um Paradigma Possível para Elicitação de Requisitos de Software*, MARTINS [26] procura relacionar a Teoria da Atividade, um conceito oriundo da Psicologia, para melhor orientar a atividade de elicitação de requisitos de software.

De nossa parte, o esforço de fomentar o diálogo entre saberes esteve presente na própria constituição do Comitê de Programa do WOSSES 2006 e do WOSSES 2007. Quando convidávamos alguém de fora da ES, invariavelmente esbarrávamos na perplexidade do convidado: ‘mas como é que vou avaliar um artigo da ES se não sou engenheiro de software?’. Explicávamos que se tratava de um experimento desafiador, de resultados imprevisíveis, mas que valeria a pena ser tentado, até por conta dos próprios objetivos do WOSSES, o de quebrar velhas e rígidas fronteiras disciplinares. O Comitê foi constituído com um empresário, três membros de órgãos do governo, e professores/pesquisadores assim distribuídos por especialidade: dez oriundos da ciência da computação, um da engenharia de produção, um da administração, um da psicologia, três da sociologia, um da educação, um da antropologia e um da comunicação. Seus trabalhos foram muito bem sucedidos, mesmo quando avaliações controversas exigiram um debate entre diversos especialistas reunidos em torno de um mesmo texto de ES.

Em recente conferência, BOEHM [8], a partir da definição de ES oferecida por um dicionário - “*aplicação da ciência e da matemática pelas quais as propriedades do software tornam-se úteis às pessoas*” – procurou esclarecer a respeito dos saberes envolvidos que:

serem ‘úteis às pessoas’ implica que é forçoso incluir as ciências do comportamento, da gestão e da economia entre as ciências relevantes, assim como a própria ciência da computação [...] o software e sua engenharia serão essenciais para o sucesso, mas também serão críticas as novas competências para integrar a engenharia de software à engenharia de sistemas, ao marketing, às ciências contábeis e às competências próprias ao núcleo central da ciência da computação.

Também recentemente, HERBSLEB [4, p.25] lança mais luzes sobre os limites disciplinares da ES:

[...] a arquitetura do produto e a estrutura organizacional são intimamente relacionadas [...]. Se assumimos que podemos projetar as arquiteturas em bases puramente técnicas, colocamos em risco nossas organizações e nossos clientes, mas, por ora, compreendemos relativamente pouco sobre como pensar a respeito deste problema [...]. Precisamos da pesquisa interdisciplinar para compreender as restrições que as arquiteturas impõem às organizações, e que as organizações impõem às arquiteturas, para entender como estruturas técnicas e organizacionais podem co-evoluir.

Com esta definição, o autor oferece uma versão muito interessante do que vem a ser o olhar sociotécnico. Um olhar apuradamente percebido por Tracy Kidder, em seu livro *The soul of a new machine* (apud LATOUR [9]), quando conta a história de construção da máquina Eclipse da Data General, concorrente do VAX da DEC (Digital Equipment Corp.), no final dos anos 70. Uma história que começa com Tom West, chefe da equipe de desenvolvimento do que ainda se chamava internamente de Eagle, e que vinha a ser uma máquina de 32 bits para competir com os VAX de 32 bits da DEC. Para ter uma idéia de como era a máquina do concorrente, West arranhou uma visita às escondidas a uma empresa que possuía o VAX então recém lançado:

(em um dia qualquer de 1978) ao examinar o VAX, pareceu a Tom West estar diante de um organograma da DEC. Sentiu que o VAX era muito complicado. Ele não gostou, por exemplo, do sistema em que as várias partes da máquina se intercomunicavam; para o seu gosto, havia muito protocolo envolvido. West concluiu que o VAX encarnava as falhas da organização corporativa da DEC. A máquina expressava o sucesso fenomenal do estilo burocrático e cauteloso da companhia.

Eis um exemplo preciso do que seja o olhar sociotécnico, aquele que vê o técnico e o social em uma mesma mirada, um olhar sinótico, como aquele que anima a descrição de Tracy Kidder, os argumentos de James Herbsleb, e a própria exposição de Bruno Latour em seu livro *Ciência em Ação*, no qual expõe um atributo fundamental do olhar sociotécnico, a saber: “*contexto e conteúdo se confundem*”.

Retornemos, pois, à questão: como navegar pela complexidade? SUCHMAN [10], em seu livro *Plans and situated actions*, oferece valiosas respostas a partir das diferenças entre *planos* e *ações situadas*, denominações constituídas pelas pesquisas do antropólogo Thomas Gladwin, publicadas em 1964, sobre pescadores da Micronésia. Ele chamou a atenção para as técnicas de navegação de um daqueles povos, os truqueses, utilizadas para viagens longas em mar aberto, apontando especialmente os contrastes entre a sua maneira de navegar e aquela dos europeus. Para os propósitos deste artigo, a saber, de discutir as práticas da ES, é perfeitamente cabível aproximar a noção de plano à de modelo, e é por esta razão que, ao referir as questões de Lucy Suchman e Thomas Gladwin, passamos a utilizar ambas as noções como sinônimas, reunindo-as sob a denominação plano/modelo.

Segundo Gladwin, o navegador europeu principia com um plano/modelo – um curso – projetado de acordo com certos princípios universais, ao qual relaciona todos os movimentos de sua viagem, e, portanto, seu esforço é o de manter-se no curso previamente planejado. Se eventos inesperados ocorrem, o navegador europeu tem de primeiro alterar o plano/modelo para somente então responder devidamente. Já o navegador da Micronésia começa com um objetivo em vez de um plano/modelo, partindo rumo ao seu objetivo e respondendo às eventualidades, de forma *ad hoc*, à medida que vão aparecendo. Desta forma, faz pleno uso das informações fornecidas pelo vento, pelas ondas, pela maré, pela corrente, pela fauna, pelas estrelas, pelas nuvens, pelo som da água batendo no barco, navegando em absoluta conformidade com todas elas. Seu esforço é direcionado para alcançar seu objetivo, porém, se lhe é fácil responder sobre seu objetivo, não consegue fazê-lo com relação ao seu curso.

Esse esforço de partir rumo a um objetivo de forma *ad hoc* constitui aquilo que Suchman denomina de uma *ação situada*.

Para SUCHMAN, [10, p. ix], planos/modelos revelam-se como um recurso fraco frente a atividades primariamente *ad hoc*. Em verdade, dado o viés europeu de nossa cultura, só quando somos pressionados para prestar contas da racionalidade de nossos atos é que invocamos um plano/modelo como guia. Previamente proposto, planos/modelos são necessariamente vagos, na medida em que tem de acomodar as contingências imprevisíveis de situações sempre particulares. Reconstruído em retrospecto, o plano/modelo filtra precisamente as especificidades dos detalhes que caracterizam a ação situada, em favor somente daquelas ações que podem ser enquadradas por sua eventual conformidade ao plano/modelo. O artigo *Por uma perspectiva sociotécnica no desenvolvimento de sistemas de computação: o exemplo do Modelo Mikropolis*, apresentado ao WOSSES em 2006 por PORTO DE ALBUQUERQUE [11], também ressalta a filtragem do real quando se trata da formalização de um plano/modelo, tratando-a como uma *descontextualização* [11]:

A perspectiva sociotécnica [...] compreende o processo de transformação de padrões de ação social para o formato de informação técnica, a qual entra novamente no espaço social, modificando-o. A primeira etapa desse processo exige uma **descontextualização** do padrão de ação social estabelecido, para o qual só assim se pode construir uma versão formal, por exemplo, na forma de um artefato tecnológico de software [...].

Na visão planejadora / modeladora, planos e modelos são pré-requisitos para a ação, prescrevendo-a em todo o detalhe. Porém, o curso da ação somente pode ser projetado ou reconstruído em termos das intenções prévias e situações típicas (as “melhores práticas”) visto que o significado prescritivo das intenções frente à ação situada são inerentemente vagos. A coerência da ação situada está vinculada não a predisposições individuais ou a regras convencionais, mas sim a interações locais contingentes, de acordo com as circunstâncias particulares em que se encontram os atores. Desta forma, rejeitam-se soluções “universais”, as quais, trazidas para o campo da ES conforme argumento de TEIXEIRA [12] em seu artigo *Algumas observações sobre os vínculos entre a Engenharia de Software e o pensamento moderno*, apresentado ao WOSSES em 2006, constituem o que chama de herança do pensamento moderno:

Com Descartes, a racionalidade científica ocidental passou a valorizar o método como diretriz para a verdade. Isso se entranhou na matemática e nas ciências naturais, que, de sua parte, influenciaram o estabelecimento daquilo que conta como conhecimento no ocidente [13]. A construção de modelos universalizantes, representativos de sistemas ordenados, é uma das idéias básicas do pensamento moderno e, sob esse estatuto, achamos natural que o desenvolvimento de sistemas de software se ampare nas idéias de representação, formalização, ordem, controle, sofrendo profunda influência do racionalismo, da metáfora mecanicista e da valorização do método [14]. Assim, o conceito de software como um produto de engenharia, como um objeto matemático, encontrou livre curso nos primeiros 50 anos da história do desenvolvimento de software [3].

Resumindo, todo curso de ação depende das suas circunstâncias materiais e sociais. Em vez de abstrair a ação de suas contingências, representando-a como um plano/modelo

racional e universal, a abordagem proposta é a de estudar como os envolvidos na ação podem usar as suas circunstâncias para alcançar o que se pode chamar de uma *ação inteligente*. Segundo SUCHMAN [10],

em vez de construir uma teoria da ação como produto de uma teoria de planos, de modelos, o objetivo é investigar como as pessoas produzem e encontram evidências para seus planos no curso da ação situada. De forma geral, mais que subsumir os detalhes de uma ação no estudo dos planos, os planos são subsumidos no problema maior da ação situada.

Assim, planos e modelos devem ceder sua primazia aos ditames da ação situada. SILVA & LIMA [15], em *Análise de Requisitos de Software e Análise da Atividade de Trabalho*, artigo apresentado ao WOSSES 2005, tocam nesta mesma questão pelo viés do conceito de *análise da atividade*, oriundo da ergonomia:

[...] pode-se compreender a relação existente entre essa metodologia (análise da atividade) e a desenvolvimento de softwares. A relação passa pela compreensão da atividade de trabalho, de modo que aspectos determinantes da atividade real de trabalho sejam incorporados (ou pelo menos considerados), na concepção da ferramenta computacional. O que comumente serve de base para os analistas de sistemas não é um **modelo da atividade real**, mas sim um **modelo da tarefa prescrita**, muitas vezes elaborado sem a participação direta do usuário final.

Se, em termos do que propõe Suchman, fica complicado imaginar o que seria um “modelo da atividade real”, não é difícil, todavia perceber a natureza idêntica do problema apontado por esses autores, e, portanto, a forte correlação existente entre os conceitos de atividade real e de ação situada, assim como entre o de tarefa prescrita e o de plano/modelo.

3 Resumindo Algumas Características Aproximativas do Olhar Sociotécnico

Se engenheiros de software nem de longe se parecem a pescadores da Micronésia, não é difícil, todavia, compreender, a partir de experiência tão diversa, os desafios que enfrentam diante do que se caracteriza claramente como um problema de construção do conhecimento. Uma certa fetichização de modelos por parte desses engenheiros inverte, na prática, a metodologia proposta por Suchman, – a da primazia da ação situada sobre os planos/modelos. Portanto, caso se admita para a engenharia de software um olhar sociotécnico, ela terá de abrir mão das suas obsessões por modelos universalizantes para debruçar-se sobre a tarefa bem mais desafiadora de enfrentar as especificidades irredutivelmente locais de toda ação situada. Para dar conta de tais particularidades, ou dito de outra forma, para relacionar modelos definidos aprioristicamente com contingências locais, será preciso lançar mão de um olhar que alcance além das reduções matematizantes, algorítmicas e simplificadoras empreendidas pelos modelos, a saber, um olhar que dê conta das indivisíveis imbricações do pano sem costura que perfaz o mundo real. Um olhar sociotécnico, que alcance de um só golpe, sinoticamente, o técnico e o social, terá de ser necessariamente um olhar interdisciplinar.

Resumindo de forma breve e fazendo uso de esquemas dualistas (necessariamente simplistas, porém de razoável eficiência comunicativa), pode-se dizer que para o olhar sociotécnico prevalecem todas as características descritas nos itens subseqüentes desta seção.

3.1 O local, o situado (resistência ao global, ao universal), o caso a caso, a contingência

BOEHM [8], a exemplo do já citado artigo de TEIXEIRA [12], distingue a preocupação com o local como contraponto obrigatório ao legado modernista:

A teoria subjacente aos modelos de processo de software tem de evoluir das visões de mundo ‘modernas’ – puramente reducionistas – (universal, generalizante, atemporal, escrita) para uma síntese entre estas visões e as visões de mundo ‘pós-modernas’ – situadas – (particular, local, temporal, oral).

A insistência modernista em modelos capazes de apreender o real em sua complexidade só pode subsistir sob a hipótese de um mundo estável e regular, quando não um mundo feito de causalidades determinísticas. Retornamos ao artigo de TEIXEIRA no qual se pode encontrar uma abordagem apropriada do problema:

Vejamos a *modelagem de informação*, ferramenta chave com presença muito forte nos ciclos de desenvolvimento de sistemas. Ela denota “uma posição realista ingênua”, que pressupõe bastar existir um método adequado para que o mundo real, objetivo, possa ser descoberto. Mais ainda, se descoberto através desse método, tal mundo seria consistente, de complexidade gerenciável e, portanto, controlável. A pressuposição subjacente é que existe um mundo ordenado *a priori*, bastando ao engenheiro de software descobrir / capturar os requisitos preexistentes, formalizar uma especificação e desenvolver o sistema desejado a partir dela. A maioria das abordagens tende a considerar que é possível, de antemão, definir os requisitos e que eles se manterão estáveis ao longo do desenvolvimento [...] De posse da representação, os processos de desenvolvimento assumem uma estrutura hierárquica, com uma forte crença no poder do projetista em deter todo o controle da situação [16]. Deixam de considerar o mundo real e passam a focalizar apenas a representação do sistema [14].

Retomando as idéias de Barry Boehm, é ele mesmo quem cita suas pesquisas, realizadas em conjunto com M. Phongpaibul, a respeito da implantação de processos de melhoria de software na Tailândia a partir da adoção do CMM (*Capability Maturity Model*) naquele país [17]. Através da utilização de dimensões comparativas que denominam de culturais, alinhadas ao longo de atributos tais como individualismo/coletivismo, masculinidade/feminilidade, aversão a incertezas, orientação para o longo prazo, hierarquias de poder, entre outros, BOEHM & PHONGPAIBUL [17] propõem que tais dimensões freqüentemente explicam as diferenças na adoção de produtos e processos de software quando certas culturas são cotejadas lado a lado. O exemplo que oferecem é o da baixa taxa de adoção do CMM em organizações tailandesas (17 de 380 usuários experimentais), sendo este um modelo constituído a partir de uma cultura como a norte-americana, mais individual, masculina, e voltada para o curto prazo quando comparada à cultura tailandesa, marcadamente mais coletiva, feminina e voltada para o longo prazo.

Seguindo no exemplo da implantação de CMM, o desafio não é apenas o de implantar um modelo universal, mas também o de projetar relações, papéis e habilidades que se esperam dos desenvolvedores. Ainda que seja matéria típica de projeto (*design*) a elaboração das relações entre o universal – presente nos modelos, padrões e métodos – e o local – a utilização prática, situada, desses modelos, padrões e métodos [16, p.135] –, modelos e padrões usualmente focalizam o problema sob um viés tecnicista simplificador, desprezando questões culturais, sociais e políticas. Por isso mesmo, quando a implantação de um modelo não é bem sucedida, essas questões freqüentemente surgem como explicação para o fracasso, mantendo a aura de infalibilidade “técnica” do modelo, assim preservando sua presumida competência. Porém, por si só, nenhum modelo ou padrão pode garantir a repetição de um suposto sucesso obtido em alguma situação anterior; em verdade, ele replica apenas a si mesmo sob a alegação de replicar a competência que lhe é “intrínseca”. Para que venha a ser utilizado de fato, um modelo ou padrão universal, em última instância, terá sempre de elaborar seu significado localmente. Em seu artigo *Helping the software development community collaborate*, apresentado ao WOSSES 2006, ARAÚJO et al [18] destacam a respeito que “a adoção de qualquer modelo de referência ou de prática de ES requer uma mudança cultural tanto dos profissionais de software quanto dos níveis gerencial e executivo das organizações de software”. Trata-se, portanto, de um problema que só se resolve plena e satisfatoriamente no caso a caso, como denota a afirmação:

Embora as áreas de processos descrevam comportamentos que devem ser exibidos em qualquer organização, práticas devem ser interpretadas utilizando um profundo conhecimento de CMMI, da(s) disciplina(s), da organização, do ambiente de negócios e das circunstâncias específicas envolvidas. [19, p.97, grifo nosso].

3.2 A complexidade (em vez de simplificações)

Hoje em dia é cada vez mais familiar a premissa conceitual que interpreta os Sistemas de Informações (SI) como sistemas sociotécnicos [20]. Portanto, reconhecer a complexidade sociotécnica do objeto da ES torna mais fácil aceitar que gerir um projeto de software (desenvolvimento, manutenção ou implantação de um processo), desde sua concepção até sua implantação, deve ser visto como gerir não somente o desenvolvimento de código e da infra-estrutura de redes e computadores como também, por exemplo, os interesses dos diversos grupos e pessoas [21]. Assim, dá-se o que se poderia chamar de um processo de negociação sociotécnica, no qual o trabalho do engenheiro de software, na prática, necessariamente é o de urdir redes sociotécnicas pelas quais negocia os papéis de entidades indistintamente “sociais” e “técnicas”.

No entanto, historicamente, os projetos de software têm sido vistos apenas como a tarefa de especificar (e seguir) padrões técnicos, deixando de lado a complexa, e por vezes incontrolável, tarefa de alinhar inovações, tecnologias, cultura, políticas, condições de mercado e organizacionais [16, pp.8-9]. É comum a idéia, baseada numa postura realista, segundo a qual existe um mundo exterior ordenado e, conseqüentemente, que é possível ao engenheiro de software descobrir / capturar os requisitos que de alguma forma são dados de

antemão. Em seguida, especificações formalizadas em vários níveis de abstração servem como guia hierárquico de todo esforço de construção dos sistemas de software, sob uma confortadora e tradicional premissa pela qual se crê que o projetista detém, *a priori*, o controle total do processo. O olhar sociotécnico, ao não dividir *a priori* a complexidade do objeto da ES em aspectos técnicos e não-técnicos, ou humanos e não-humanos, reconhece a exposição do projetista/engenheiro de software ao contingente, ao local, ao situado. Desta forma, somos instigados a detalhar a percepção e a descrição dos mecanismos concretos que viabilizam a coesão das redes sociotécnicas que garantem a existência e o sucesso de softwares, de processos de softwares, de métodos, técnicas e tecnologias de desenvolvimento. Trata-se de uma visão onde modelos e especificações são entidades, como diversas outras, que devem ser convenientemente enredadas na rede que se quer estabilizar.

3.3 Conhecimentos não formalizáveis

A prática científica não se constitui somente em seguir regras universais. Consiste mais especificamente de cursos particulares de ação, completamente compreensíveis apenas no seu contexto local (a depender inclusive das pessoas que estão envolvidas nessas práticas, na medida em que sempre há um nível pessoal de realização). Universalidade e independência de contexto não podem ser tomadas como dadas, mas devem ser analisadas como realizações (conquistas) precárias – por exemplo, como resultado da construção bem sucedida da expansão de redes conectando humanos e não humanos.

Uma versão desta rejeição ao universalismo pode ser dada pelo contraste entre conhecimento explícito e conhecimento tácito. O primeiro é feito de informações ou instruções que podem ser formuladas em palavras ou símbolos e que, portanto, podem ser armazenadas, copiadas e transferidas por meios impessoais, tais como documentos ou arquivos de computador. O conhecimento tácito, por sua vez, é conhecimento que não foi (até porque não pôde ser) formulado completa e explicitamente e, portanto, não pode ser efetivamente armazenado e transmitido inteiramente por meios impessoais. As habilidades motoras oferecem bons exemplos a respeito: sabemos dirigir uma bicicleta mas dificilmente saberíamos colocar tal habilidade em palavras. Da mesma forma, ao ensiná-lo aos nossos filhos, evitamos livros e manuais e partimos para o ensino pessoal, procurando mostrar-lhes como se faz.

Embora a importância do conhecimento tácito seja amplamente reconhecida em inúmeras atividades humanas, só é possível pensar em sua transmissão através do contato direto entre o aprendiz e aquele que já sabe, ou seja, trata-se de um conhecimento que não pode ser encapsulado, por exemplo, pela Inteligência Artificial. O foco no *método* que acompanha a prática tradicional e universalista da tecnociência não dá a devida importância ao conhecimento tácito, embora este último seja crucial para as práticas da tecnociência. A

atenção ao conhecimento tácito explicita que a tecnociência não é mera e simplesmente um empreendimento cumulativo que resulta em avanços permanentes⁶.

3.4 Os transbordamentos (em vez de enquadramentos)

Os planos/modelos fazem necessariamente uma série de suposições a respeito do mundo sobre o qual se propõem a intervir. Tais suposições dizem respeito à representação que fazem desse mundo, ou seja, correspondem a um determinado recorte que operam nesse mesmo mundo. Ao recortá-lo, produzem uma simplificação, ou dito de outra forma, uma redução da complexidade, sem a qual não teriam como adquirir “generalidade”, “universalidade”. Chamamos a esta operação de recorte de enquadramento, pelo qual se destaca do mundo aquilo que deve ser levado em conta, ou seja, aquilo que, por pertencer ao quadro, tem de ser levado em consideração. Todavia, ao fazê-lo, se estabelece, ao mesmo tempo, tudo aquilo que fica “de fora” do quadro e que, portanto, não pertence ao mundo sobre o qual os planos/modelos intervêm⁷. A tudo que fica “de fora” por conta de um enquadramento, chamamos de transbordamento [25].

A questão que se coloca é que, a todo enquadramento, corresponde um transbordamento. Dito de outra maneira, se alguma forma de enfrentar a complexidade é necessária para que diante dela não se sucumba (e por isso há sempre algum nível de enquadramento), todavia não é ela apreensível de um só golpe (e por isso há sempre transbordamentos). Se a capacidade de enquadramento é uma medida de sucesso de um plano/modelo, o transbordamento indica a resistência que se lhe opõe. Porém, é através dos transbordamentos que se pode conhecer melhor a que mundo se refere o plano/modelo, e, portanto, verificar sua pertinência ao plano/modelo aplicado.

Em termos de ES, os modelos fazem várias suposições: organizacionais, culturais, sociais, ambientais, arquitetônicas, de relações de trabalho, para mencionar alguns exemplos. Tais suposições são “naturalizadas” na medida em que se torne mais corriqueira a prática de forçar a adequação dos recursos (*hardware*, *software*, engenheiros, etc.) de uma entidade

⁶ A distinção entre conhecimento tácito e codificado foi introduzida por Michael Polanyi, especialmente em seu livro *The Tacit Dimension* (1987). MACKENZIE [24] assume uma proposta instigante a respeito do conhecimento tácito ao tratar da bomba atômica. Sua proposta, a da “desinvenção” da bomba atômica, emerge da idéia segundo a qual, a partir do momento que um conhecimento relevante pudesse ser desaprendido, as armas atômicas a ele ligadas seriam “desinventadas”. Bastaria que em seu projeto e produção houvesse um hiato suficiente para a desapareição de todo o seu conhecimento tácito. Ainda assim, as armas atômicas poderiam ser reinventadas, mas não simplesmente copiando artefatos, diagramas e instruções explícitas. Em certo sentido, elas teriam de ser reinventadas.

⁷ O artigo “Por que Falham os Projetos de Implantação de Processos de Software?”, de TEIXEIRA & CUKIERMAN [28], apresentado ao WOSSES em 2007, propõe uma discussão mais detalhada sobre o que fica “de fora” do quadro da ES.

(corporações, instituições públicas, pequenas empresas, etc.) ao modelo proposto, salvaguardando-o de qualquer “culpa”, a qual deve caber única e exclusivamente a uma adequação mal realizada. No limite, pode-se dizer que “culpa-se” o mundo e “absolve-se” o modelo, a saber, mantém-se intacto o enquadramento por ele realizado. Outra opção, a que propomos, é a de “absolver” o mundo e assim poder aprender mais sobre a aplicabilidade do modelo através de seus transbordamentos. Se o sucesso de um modelo fala bem sobre a excelência de seus pressupostos, seu fracasso fala mais alto sobre algo bem mais interessante, o mundo em que vivemos.

Concluindo, ao propor a indissociabilidade do “técnico” e do “social”, do “conteúdo” e do “contexto”, o olhar sociotécnico opõe ao “desejo normativo” – o de simplificar; de instituir a norma, o modelo; de planejar; de universalizar; de produzir similaridades – o “desejo descritivo” – o de descrever em detalhes; de particularizar; de localizar; de especificar; de produzir diferenças. Vale lembrar que, ao optar pela localidade e pelas especificidades, tem-se também um desejo brasileiro, o de recuperar nossas circunstâncias, e assim estabelecer uma trégua com circunstâncias que são eminentemente alienígenas.

4 Por onde prosseguir? Primeiras especulações sobre possíveis caminhos

Na contramão das “melhores práticas” dos artigos de ES, chegamos ao momento das conclusões sem qualquer “proposta concreta”, sem nenhuma “proposta de solução” em mãos. Todavia, é HERBSLEB [4, p.26, grifo nosso] quem, com muita propriedade, alerta quanto ao que qualifica de

forte e desafortunada tendência em ES de pensar que todo artigo deveria exibir um resultado prático que seja imediatamente útil [...]. Penso que é bastante razoável esperar que *programas* de pesquisa tenham de conduzir a resultados práticos, mas temos de despende mais tempo e energia compreendendo como as coisas funcionam.

Dito de outra forma, o presente artigo tem única e exclusivamente a pretensão, que já não é pouca, de problematizar as práticas correntes da ES, pelas quais os “fatores não técnicos” são “expulsos” do seu alcance disciplinar, ou então, quando muito, tratados como “aspectos” de segunda linha, sempre subordinados a enquadramentos “técnicos”. Assim, não vemos alternativa a não ser pugnar por uma maior aproximação com as ciências humanas e sociais, materializando um movimento necessariamente interdisciplinar que, a nosso ver, é a única saída para enfrentar questões prementes do desenvolvimento de software com qualidade para os quais a ES, se preservada em sua “exclusividade” disciplinar, não tem como abordar.

Neste sentido, nossas propostas dizem respeito não a esta ou aquela “solução concreta”, mas à própria reconfiguração da agenda de pesquisas em ES de forma a incluir

novos instrumentos teóricos e metodológicos advindos de outras áreas do conhecimento. A partir da antropologia e da história, propomos que as pesquisas em ES venham a privilegiar:

4.1 As “descrições densas”

Em seu primeiro capítulo de *A Interpretação das Culturas*, o antropólogo Clifford Geertz discute o trabalho do etnógrafo. Resumindo-a brevemente, sua posição é a de que o objetivo do etnógrafo deve ser o de observar, registrar e analisar uma determinada cultura. Mais especificamente, que ele/a deve dedicar-se à interpretação de signos de forma a alcançar seus significados em meio à cultura em questão. Tal interpretação deve estar baseada no que denomina de *descrição densa* de um signo, pela qual se torna possível apreender todos os seus sentidos. Seu exemplo [22, pp. 6-7] sobre a “piscadela de um olho” esclarece o ponto. Quando alguém pisca um olho, está apenas “contraíndo rapidamente suas pálpebras”, como constaria em uma *descrição superficial*, ou estaria, por exemplo, “imitando uma piscadela para brincar com alguém, fazendo-o acreditar em uma conspiração em marcha”, caracterização somente possível através de uma *descrição densa*? É nas diferenças entre ambas as descrições que reside o objeto da etnografia: uma

hierarquia estratificada de estruturas de significados em termos da qual tiques nervosos, piscadelas, falsas piscadelas, brincadeiras de piscar de olho [...] são produzidos, percebidos e interpretados, e sem a qual de fato não existiriam, não importando o que alguém fizesse ou não com suas pálpebras [22, p.7].

Ainda segundo suas palavras, [22, pp.9-10]

[...] etnografia é descrição densa. O que o etnógrafo enfrenta de fato [...] é a multiplicidade de estruturas conceituais complexas, muitas delas sobrepostas ou amarradas umas às outras, que são simultaneamente estranhas, irregulares e implícitas, e que ele tem que esquematizar de alguma forma, primeiramente para apreendê-las e em seguida representá-las. [...] Fazer etnografia é como tentar ler (no sentido de ‘construir uma leitura de’) um manuscrito – estranho, desbotado, repleto de elipses, incoerências, emendas suspeitas e comentários tendenciosos, mas escrito não com os sinais convencionais do som, mas com exemplos transitórios de comportamento moldado.

Através da *descrição densa*, Geertz espera que a compreensão mais detalhada dos signos estabeleça ou amplie o diálogo entre culturas diversas, conforme se pode depreender da passagem na qual afirma que “*compreender a cultura de um povo expõe sua normalidade sem que se reduza sua particularidade. [...] Torna-o acessível: ao enquadrá-lo em sua própria banalidade, dissolve-se sua opacidade*” [22, p.14]. Para o caso da ES, podemos entender, em uma primeira instância, que o diálogo a ser estabelecido ou ampliado é aquele entre as culturas dos que vão estudar a implantação dos modelos/planos de desenvolvimento de software (o etnógrafo da ES), dos que os adotam em seu cotidiano profissional (os profissionais), e dos que os concebem e difundem como sendo as melhores práticas em ES (os pesquisadores, professores e consultores). Porém, há de se destacar uma segunda instância, o diálogo em meio às diferenças entre as culturas dos produtores de modelos/planos (majoritariamente norte-americanos) e a de seus consumidores, entre eles,

nós, os engenheiros brasileiros. Uma descrição densa aplicada à ES não só tem a capacidade de elucidar em que de fato consiste a prática da ES como também quais são as tensões e assimetrias decorrentes da adoção, em nossas instituições e corporações, de modelos/planos que não foram originalmente concebidos para atender às suas particularidades e especificidades. Descrever densamente a prática do desenvolvimento de software em nosso país é, a nosso ver, um caminho imprescindível para que se possa conceber a serventia e utilidade de uma engenharia de software brasileira, não somente por conta da nacionalidade de seus quadros, mas, e principalmente, pela sua capacidade de desenvolver um conhecimento em ES adequado às (e problematizado a partir das) necessidades locais.

Tantos diálogos só adquirem sentido caso se queira, parafraseando GEERTZ [22, p.5] compreender o que é a ES, tarefa a exigir toda atenção, em primeira instância, não às suas teorias ou suas descobertas, e certamente não àquilo que seus apologistas dizem a respeito dela, mas sim ao que fazem seus praticantes.

4.2 A “desnaturalização” dos modelos e artefatos através de suas histórias

A análise histórica de um modelo/processo de desenvolvimento de ES, além de oferecer lições para o desenvolvimento de novas tecnologias de produção de software [11], é ela mesma uma contextualização indispensável para que se compreendam quais os pressupostos e o alcance das promessas de um determinado modelo/processo. Um modelo/processo sem história torna-se um “universal”, fazendo supor que sua aplicação pode ser feita da mesma forma, e com os mesmos efeitos, a qualquer tempo e em qualquer lugar. Desta forma, uma solução, se isolada das circunstâncias históricas de sua concepção – quais os problemas originalmente enfrentados, quais os efeitos então pretendidos, quais os beneficiados, etc. – acaba tornando-se uma “solução natural”. Nossa proposta é caminhar par e passo com a história das soluções propostas para a ES, ou seja, “desnaturalizá-las”, procurando, através de sua historicidade, estabelecer parâmetros que permitam avaliar suas circunstâncias de origem face às efetivas circunstâncias de seu uso.

No caso da ES, há uma particularidade de excepcional relevância, caso se aceite o argumento de MAHONEY [23]. Isto porque, em seu artigo *Finding a History for Software Engineering*, [23, pp.8-9], historiadores e os próprios engenheiros são surpreendentemente colocados lado a lado no interesse pela história da engenharia de software. A princípio, ambos estariam caminhando por vias paralelas, com os historiadores observando “de longe” os movimentos dos engenheiros em busca da constituição do que seria a engenharia de software como um campo do conhecimento:

a questão de interesse para os historiadores seria portanto como os engenheiros de software têm enfrentado a tarefa da auto-definição. Em larga medida, abordar essa questão implica em observar e analisar as respostas que os profissionais têm a oferecer [...] porém, em vez de postular um possível consenso entre profissionais a respeito da natureza da engenharia de software, o que os historiadores podem fazer é perseguir os esforços para se alcançar um tal consenso.

Como a história da ES é muito recente, e, portanto, tem ela mesma muito a contribuir para iluminar os caminhos ainda a percorrer, Mahoney supõe que as vias pelas quais caminham engenheiros de software e historiadores não são assim tão paralelas e distantes. Ao contrário, chega mesmo a apontar uma convergência de interesses:

Pode ser de alguma valia pensar em historiadores e engenheiros engajados em uma causa em comum. Ambos procuram por uma história para a engenharia de software, ainda que não com os mesmos propósitos nem com os mesmos pontos de vista.

Ao leitor mais paciente e curioso que tenha nos honrado com sua leitura atenta até estas derradeiras linhas, talvez ainda reste uma última pergunta: afinal, estou mesmo diante de um artigo sobre ES? Para respondê-lo, oferecemos como nossas últimas palavras as providenciais ponderações de James Herbsleb:

A pesquisa interdisciplinar pode parecer completamente inaceitável. Em alguns ambientes, tenho ouvido algumas pessoas perguntarem a respeito de um ou outro programa de pesquisa, ‘mas trata-se realmente de ciência da computação?’. Aparentemente, não são poucas as pessoas que gastam tempo e energia significativos preocupando-se com uma questão como esta. A melhor resposta que ouvi foi a de um colega, Randy Pausch, codiretor do *Entertainment Technology Center* da Escola de Ciência da Computação da Universidade Carnegie Mellon. Quando um estudante perguntou-lhe se o projeto ao qual ele, o estudante, estava se dedicando era “realmente” de ciência da computação, Randy respondeu com uma ponta de impaciência: ‘Faça algo fantástico, depois decidimos como deveremos chamá-lo’.

Referências

1. FUGGETTA, A., 2000, “Software Process: A Roadmap”. In: FINKELSTEIN, A. (ed.), *The Future of Software Engineering*.
2. SOMMERVILLE, I., 2004, *Software engineering*. 7th ed., Addison-Wesley.
3. LEVESON, N. G., 1997, “Software Engineering: Stretching the Limits of Complexity”. In: *Communications of the ACM*, v. 40, n. 2, pp.129-131.
4. HERBSLEB, J. D., 2005, “Beyond Computer Science”. In: *Proceedings of the 27th International Conference on Software Engineering (ICSE)*, St. Louis, Missouri, EUA, pp. 23-27.
5. PRIKLADNICKI, R.; AUDY, J. L. N., 2005, “Os Aspectos Não-Técnicos Intervenientes no Desenvolvimento Distribuído de Software”, In: *Workshop Um Olhar Sociotécnico sobre a Engenharia de Software (WOSES)*, 1., Rio de Janeiro. Anais... pp. 45-55.
6. PRIKLADNICKI, R.; AUDY, J. L. N., 2006, “Construção do Conhecimento e Complexidade na Área de Engenharia de Software”, In: *Workshop Um Olhar Sociotécnico sobre a Engenharia de Software (WOSES)*, 2., Vila Velha. Anais... pp. 51-63.

7. MENDONÇA, R .M., 2006, “Usabilidade de Processos”, In: *Workshop Um Olhar Sociotécnico sobre a Engenharia de Software (WOSES)*, 2., Vila Velha. Anais... pp. 13-25.
8. BOEHM, B., 2006, “A View of 20th and 21st Century Software Engineering”. In: *Proceedings of the 28th International Conference on Software Engineering (ICSE)*, Shanghai, China, pp.12-29.
9. LATOUR, B., 2000, *Ciência em Ação: como seguir cientistas e engenheiros sociedade afora*. São Paulo, Editora UNESP.
10. SUCHMAN, L., 1987, *Plans and Situated Actions: the problem of human machine communication*. Cambridge University Press.
11. PORTO DE ALBUQUERQUE, J., 2006, “Por uma Perspectiva Sociotécnica no Desenvolvimento de Sistemas de Computação: o exemplo do Modelo Mikropolis”, In: *Workshop Um Olhar Sociotécnico sobre a Engenharia de Software (WOSES)*, 2., Vila Velha. Anais... pp. 1-12.
12. TEIXEIRA, C. A. N., 2006, “Algumas observações sobre os vínculos entre a Engenharia de Software e o pensamento moderno.”, In: *Workshop Um Olhar Sociotécnico sobre a Engenharia de Software (WOSES)*, 2., Vila Velha. Anais... pp. 39-50.
13. HIRSCHHEIM, R.; HEINZ, K. K.; LYYTINEN, K., 1995, *Information Systems Development and Data Modeling: Conceptual and Philosophical Foundations*. Cambridge University Press.
14. DAHLBOM, B., MATHIASSEN, L., 1993, *Computers in Context: The Philosophy and Practice of Systems Design*. Oxford, NCC Blackwell.
15. SILVA, A. L.; LIMA, F. P. A., 2005, “Análise de Requisitos de Software e Análise da Atividade de Trabalho”, In: *Workshop Um Olhar Sociotécnico sobre a Engenharia de Software (WOSES)*, 1., Rio de Janeiro. Anais... pp. 31-44.
16. HANSETH, O.; MONTEIRO, E., 1998, *Understanding Information Infrastructure*. Manuscript. Disponível em <<http://heim.ifi.uio.no/~oleha/Publications/book.pdf>>. Acesso em: 01 abr. 2005.
17. PHONGPAIBUL, M., BOEHM, B., 2005, “Improving Quality Through Software Process Improvement in Thailand: Initial Analysis”. In: *Proceedings of 27th ICSE, Workshop on Software Quality*.
18. ARAUJO, R. M.; CAPPELLI, C.; SANTORO, F. M., 2006, “Helping the Software Development Community Collaborate”, In: *Workshop Um Olhar Socio-técnico sobre a Engenharia de Software (WOSES)*, 2., Vila Velha. Anais... pp. 27-37.
19. CHRISSIS, M., KONRAD, M., SHRUM, S., 2003, *CMMI: Guidelines for Process Integration and Product Improvement*. Addison-Wesley.

20. ARNOLD, M., 2003, "Systems Design Meets Habermans, Foucault and Latour". In: CLARKE, S., et. al (eds.), *Socio-Technical and Human Cognition Elements of Information Systems*. London, Information Science Publishing, pp.226-248.
21. METCALFE, M., 2003, "Concern Solving for IS Development" In: CLARKE, S., et. al (eds.), *Socio-Technical and Human Cognition Elements of Information Systems*. London, Information Science Publishing, pp.135-151.
22. GEERTZ, C., 2000, *Interpretation of Cultures*. Basic Books, New York.
23. MAHONEY, M. S., 2004, "Find a History for Software Engineering". In: IEEE Annals of the History of Computing, jan/march, pp.8-19.
24. MACKENZIE, D. A., 1998, *Knowing Machines: essays on technical change*. Massachusetts, The MIT Press.
25. CALLON, M., 1998, *The Laws of the Markets*. London, Blackwell.
26. MARTINS, L. E. G., 2007, "Teoria da Atividade: Um Paradigma Possível para Elicitação de Requisitos de Software", In: *Workshop Um Olhar Sociotécnico sobre a Engenharia de Software (WOSES)*, 3., Porto de Galinhas. Anais... pp. 13-24.
27. BRIGHAM, M., INTRONA, L. D., 2007, "Invoking politics and ethics in the design of information technology: undesigning the design". *Ethics and Information Technology* 9:1-10.
28. TEIXEIRA, C. A. N., CUKIERMAN, H., 2007, "Por que Falham os Projetos de Implantação de Processos de Software", In: *Workshop Um Olhar Sociotécnico sobre a Engenharia de Software (WOSES)*, 3., Porto de Galinhas. Anais... pp. 1-12.

ⁱ Uma síntese das discussões e publicações de três edições do workshop intitulado WOSES (artigos, chamada de trabalhos, apresentações e transcrições de debates) pode ser obtida em www.cos.ufrj.br/woses). A primeira edição foi realizada em Novembro de 2005 no BNDES/RJ, reunindo alguns pesquisadores das seguintes instituições: CCET/UNIRIO; CenPRA/MCT; COPPE/UFRJ; DEP/UFMG; e FACIN/PUCRS. A segunda edição foi realizada em junho de 2006, no SBQS (Simpósio Brasileiro de Qualidade de Software), em Vila Velha, repetindo a participação das instituições de 2005 e mais a adesão de: IC/UNICAMP; UEMG; POLI/USP; e UNIFOR. A terceira edição foi realizada em junho de 2007, também no SBQS, em Porto de Galinhas, repetindo a participação de COPPE/UFRJ, CenPRA/MCT e FACIN/PUCRS, e mais a adesão de: UECE; UFPA; UFPE; UNIMEP; UNIUBE; University of Hamburg e UVIC. Além disso, um conjunto de trabalhos aceitos nas três edições do WOSES foram estendidos (em inglês) e publicados, no segundo semestre de 2007, juntamente com um complemento da discussão apresentada neste artigo, em uma edição da revista Scientia (Unisinos, 2007/1, v. 18, n. 1).