



Defeitos de Software como Erros Humanos

Karen Sandhof, Lucia Vilela Leite Filgueiras

Departamento de Engenharia de Computação e Sistemas Digitais
Escola Politécnica da Universidade de São Paulo
Av. Prof Luciano Gualberto, 158 05508-900 São Paulo – SP - Brazil
{karen.sandhof, lucia.filgueiras}@poli.usp.br

Abstract. *This paper reports a study intended to examine the introduction of defects into software from the perspective of Human Error theory. Software development is still mostly a human activity and thus can benefit from the knowledge of Human Reliability Analysis techniques on its drive for quality.*

Resumo. *Este artigo relata um estudo destinado a analisar o processo de introdução de defeitos no software à luz da teoria do Erro Humano. Uma vez que o desenvolvimento de software ainda é uma atividade predominantemente humana, sua busca por qualidade pode se beneficiar do conhecimento das técnicas de Análise da Confiabilidade Humana.*

1. Introdução

O desenvolvimento de software, apesar do grande avanço da Engenharia de Software, ainda é fortemente dependente da atividade humana: as pessoas ainda estão envolvidas em todas as etapas do desenvolvimento de software, desde a identificação de requisitos até a homologação final do produto.

Apesar disso, os modelos seguidos pela indústria de software em sua busca pela qualidade são modelos centrados em processos. CMMI, ISO9000, ISO12207 estabelecem diversos mecanismos para a melhoria da qualidade, porém apenas consideram o ser humano quando estabelecem a necessidade de treinamento das equipes. As exceções a esta regra são os modelos P-CMM, cujo foco é o desenvolvimento humano e profissional e o PSP, cuja filosofia é centrada no indivíduo.

Nem mesmo estes modelos levam em conta o fato de que o desempenho humano é fortemente influenciado pelas condições de contexto. Um indivíduo treinado e motivado também comete erros. Embora a existência de processos permita maior índice de detecção dos defeitos de software, a causa raiz precisa ser eliminada ou pelo menos controlada para que os resultados da ação humana fiquem melhores. O conhecimento das causas raízes e de sua influência na qualidade do produto são portanto, relevantes para o Engenheiro de Software.

Em sistemas fortemente dependentes da ação humana, tais como plantas nucleares, aeronaves e outros sistemas de transporte e mais atualmente, também os sistemas médicos, a confiabilidade da operação humana é alvo de estudos cuidadosos que acompanham toda a vida do sistema. Esta preocupação decorre da certeza de que os erros humanos acontecem, por diversas causas, e que se não detectados, podem vir a causar grandes perdas.



Embora estes estudos estejam focados na operação de processos e a indústria de software seja fortemente orientada a projeto, considerou-se que a forte dependência do desempenho humano confiável justificaria um olhar da Confiabilidade Humana sobre o processo de desenvolvimento de software. Assim, projetou-se um estudo de acompanhamento do processo de desenvolvimento de software, procurando-se compreender os defeitos que foram introduzidos no desenvolvimento do produto como erros humanos. O estudo foi realizado com uma equipe de desenvolvimento de software de uma empresa, durante um período limitado de tempo. Este trabalho traz os resultados deste estudo.

2. O erro humano

A preocupação com o entendimento da natureza do erro humano decorre da necessidade de evitar, na máxima extensão possível, que a ação humana leve sistemas críticos a estados indesejáveis.

Adotando-se as definições da Teoria da Confiabilidade [Almeida Junior, 2003], o conceito de erro humano se relaciona com os conceitos de falha e disfunção. Uma **falha** (*fault*) ou **defeito** de um componente refere-se a uma condição anormal do componente, causada por problemas mecânicos ou elétricos (no caso de hardware) ou pela lógica do sistema (no caso de software). No caso humano, o desempenho é determinado por mecanismos fisiológicos e psicológicos que podem guardar uma condição anormal.

O **erro** (*error*) é a diferença entre o comportamento desejado/especificado e o real. O erro humano é uma consequência natural do processo de solução de problemas, assim como o é a própria ação correta.

Finalmente, como o elemento humano é parte de um sistema, composto ainda por hardware, software e tarefas, o erro humano pode ocasionar a **disfunção** (*failure*) do sistema, isto é, o não-cumprimento de seus requisitos.

2.1 Histórico do estudo do erro e da confiabilidade humana

Os estudos do erro humano iniciaram-se em meio aos primeiros estudos da Ergonomia, mas foram Swain e Guttman (1983) que primeiro sistematizaram a análise da confiabilidade humana para uso nos procedimentos de Avaliação Probabilística de Risco em plantas nucleares. O método desenvolvido por esses autores, denominado THERP – Technique for Human Error Prediction – baseava-se na decomposição da tarefa humana em elementos, na atribuição da probabilidade de erro humano a cada elemento, na determinação de fatores influenciadores do desempenho e da dependência entre estes elementos e finalmente, na quantificação da probabilidade total de erro na tarefa.

Acidentes relevantes aconteceram no final da década de 80, muitos deles causados pela inadequação do projeto dos sistemas às capacidades cognitivas humana. Isto motivou uma nova geração de técnicas de Análise da Confiabilidade Humana, fortemente baseadas no entendimento dos mecanismos da cognição humana.

O maior expoente desta tendência foi sem dúvida James Reason (1990), que publicou sua teoria sobre o erro humano (GEMS – Generic Error Modeling System), definindo erro humano como todas as ocasiões em que uma seqüência planejada de atividades mentais ou físicas falham em atingir o resultado pretendido e quando estas



falhas não podem ser atribuídas à intervenção do acaso. Reason mostra, em seu trabalho, que os erros humanos decorrem de uma sub-especificação cognitiva e que podem se manifestar de uma variedade de formas. Ainda, aponta quatro grandes elementos na produção de um erro: a natureza da tarefa, as circunstâncias do ambiente, o mecanismo cognitivo que gerencia o desempenho humano e a natureza do indivíduo. O trabalho de Reason apresenta uma classificação bastante abrangente dos mecanismos e da manifestação dos erros.

Outros estudos procuraram identificar a influência do contexto na atividade cognitiva humana. Hollnagel desenvolveu o CREAM – Cognitive Reliability and Error Analysis Method, método de análise da confiabilidade humana que explora a relação entre o erro, a natureza das tarefas e o contexto [Hollnagel, 1998]. Sistemas de registro e análise dos eventos de erro humano foram desenvolvidos para aperfeiçoar as análises de segurança das plantas nucleares (ATHEANA – A Technique for Human Event Analysis) [NRC, 2000] e de sistemas de controle de tráfego aéreo (HERA-JANUS – Human Error in ATM) [Isaac, 2003].

Com o objetivo de efetivar a melhoria da confiabilidade humana, os estudos recentes apontam para técnicas de investigação e ação sobre as condições de contexto que ocasionam o erro [Dekker, 2002]. Mais recente é a teoria da resiliência de Hollnagel, que advoga que mais que confiabilidade, as intervenções sobre um sistema crítico devem prover a capacidade de que ele se recupere de situações de degradação das condições de trabalho, de forma que a organização, os grupos de trabalho e indivíduos continuem exercendo sua capacidade de controle do risco. [Hollnagel, 2004]

Todos estes estudos permitem concluir que os mecanismos do erro humano são inerentes ao processo cognitivo humano e desta forma, é de se esperar que o erro humano aconteça. Como estes mecanismos são afetados pelas condições de contexto no qual as ações se dão, pode-se reduzir seus efeitos e a probabilidade de sua ocorrência pela intervenção sobre estes fatores.

2.2. Fatores influenciadores do desempenho

O trabalho de Swain e Guttman (1983) definiu o conceito de Fatores Influenciadores do Desempenho (FID) – em inglês, *Performance Shaping Factors*. Estes fatores são variáveis do ambiente que reconhecidamente podem afetar a ação humana e são tratados, no trabalho, como multiplicadores da probabilidade do erro humano. Os fatores tabulados por Swain e Guttman referem-se à operação humana em ambientes industriais. No trabalho desses autores, por exemplo, a presença de um supervisor que verifica o trabalho do operador é responsável pela redução em 10 vezes da probabilidade do erro deste operador.

No CREAM, Hollnagel definiu um conceito correlato, o de condições comuns de desempenho (CPC – Common Performance Conditions). O autor definiu 9 condições para a operação de sistemas críticos: a adequação da organização; as condições de trabalho; adequação da interface homem-sistema e apoio operacional; a disponibilidade de procedimentos; o número de objetivos a serem alcançados simultaneamente; o tempo disponível para o atendimento as metas; o grau de treinamento e experiência e a eficiência na colaboração dos envolvidos.



A importância do reconhecimento dos FID está no fato de que estes são mais facilmente controláveis através de intervenções, tais como o treinamento ou o projeto adequado dos sistemas de apoio à operação. Mantendo-se estes fatores em suas condições ótimas, permite-se a redução da probabilidade do erro.

2.3. Erro humano no processo de desenvolvimento de software

Como atividade humana, o desenvolvimento de software exige habilidades cognitivas de aprendizado, concentração, diagnóstico e tomada de decisão, e está afetado por todas as possíveis formas de manifestação do erro.

Um experimento de Nakamura (1991) realizado durante um mês com um grupo de desenvolvedores mostrou um total de 120 defeitos sendo que os fatores humanos são a maioria dentre as causas dos defeitos detectados. A Figura 1 mostra o percentual dos fatores humanos nas causas de defeitos de software.

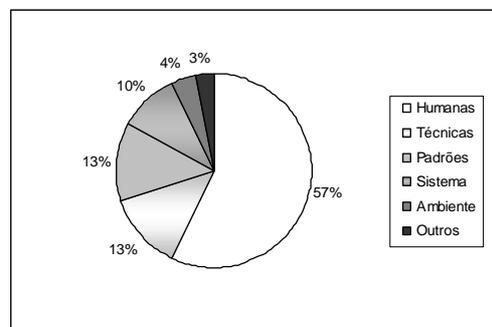


Figura 1. Causas de defeitos no experimento de Nakamura (adaptado de Nakamura et al. (1991))

Em outro estudo, Smidts e seus colaboradores [Smidts, 1998] afirmaram que falhas no processo que se traduzem em defeitos no software são resultado de erros humanos. Segundo este estudo, os erros genéricos de um analista são: omissão de requisitos, má interpretação de requisitos, falha ao notificar a existência de conflito entre dois ou mais requisitos, bem como erros topológicos, ou seja, troca na posição dos requisitos. A autora considera que erros são feitos em todos os estágios do ciclo de vida.

Stutzke e Smidt, estudando um modelo estocástico para confiabilidade de software baseado em erros humanos, afirmaram que os erros mais comumente atribuídos aos desenvolvedores de software podem ser divididos em duas categorias: “erros de desenvolvimento feitos durante análise de requisitos, projeto e codificação; e erros de depuração feitos durante tentativa de remoção de defeitos identificados durante a inspeção e testes dinâmicos de software.” [Stutzke, 2001]

Os mesmos mecanismos descritos por Reason podem afetar o processo de desenvolvimento de software. Por exemplo, é de se esperar que um desenvolvedor se esqueça de tratar um requisito durante o processo de codificação se ele for interrompido em sua concentração (a “omissão após interrupção”, de Reason); pode ser que o desenvolvedor aplique inadequadamente um padrão de projeto apenas porque ele é muito comum (a “força da regra”, de Reason). Pode ainda o erro cometido não ser



capturado se a equipe de teste estiver vinculada à equipe de desenvolvimento (a “revisão enviesada”, ou o “viés de confirmação”).

As condições descritas por Hollnagel como influenciadoras da atividade humana também estão presentes na indústria de software: a empresa pode estar inadequadamente organizada e transmitir aos desenvolvedores um ambiente de inquietação; as condições ambientais de trabalho e mesmo salariais podem não ser satisfatórias. Os sistemas de apoio ao desenvolvimento podem não dar o devido apoio aos desenvolvedores ou mesmo induzi-los erros. O suporte operacional e a disponibilidade de procedimentos podem se traduzir na existência de um processo de desenvolvimento devidamente implantado e praticado, que se inexistente, sabidamente é uma causa da baixa qualidade dos produtos de software. Um número de objetivos a serem alcançados simultaneamente e o pouco tempo disponível para o atendimento às metas são reconhecidamente problemas comuns na indústria de desenvolvimento de software. O grau de treinamento e experiência dos desenvolvedores e a eficiência na colaboração dos envolvidos, da mesma forma, são fatores que afetam a qualidade do produto de software e neste caso específico, são consideradas nos modelos de melhoria de processo.

Desta forma, observa-se que um olhar de confiabilidade humana sobre a indústria de desenvolvimento de software pode contribuir com os mecanismos empregados para a redução dos erros humanos e conseqüentemente, pode colaborar na melhoria de qualidade de software. Especificamente, espera-se que a monitoração dos FID reconhecidamente importantes e que os mecanismos de investigação e registro dos erros possam contribuir neste processo de melhoria, assim como vêm contribuindo para os processos críticos.

3. Método do estudo

Para lançar o olhar da confiabilidade humana sobre o processo de desenvolvimento de software, procurou-se estudar uma equipe de desenvolvimento durante um tempo determinado. Os objetivos do estudo foram, primeiramente, identificar FIDs reconhecidos pela equipe de desenvolvimento como tal e em seguida, estabelecer um protocolo de observação dos FIDs e dos defeitos de software que permitissem o acompanhamento de longo prazo. O estudo foi conduzido junto a uma empresa de grande porte, em setores de desenvolvimento de software destinado a soluções de automação bancária e de comércio eletrônico.

O estudo foi realizado em duas etapas. Uma pesquisa preliminar teve o objetivo de identificar quais variáveis de contexto afetavam o desempenho humano e que tipos de defeitos de software eram reconhecidos pelos desenvolvedores. A segunda etapa da pesquisa teve por objetivo o acompanhamento mais minucioso do processo de desenvolvimento e das variáveis de contexto e foi conduzida por meio de um protocolo oferecido aos desenvolvedores na forma de páginas na intranet da empresa.

O protocolo final foi seguido por 10 desenvolvedores por um período de 19 dias úteis. Este tempo não foi considerado adequado para se tirarem conclusões sobre a correlação entre os FIDs e os defeitos de software, mas foram suficientes para consolidar o protocolo de observação.



Os desenvolvedores foram indicados à equipe de pesquisa por seus superiores imediatos, considerando-se o critério de forte engajamento em atividades de melhoria da qualidade. A autorização da diretoria para a pesquisa acadêmica oficializou a tarefa atribuída aos participantes. O número reduzido de participantes trouxe o benefício de um acompanhamento individual durante todo o processo de coleta de dados. Obteve-se, ao final, 95% das respostas planejadas, o que indicou um envolvimento satisfatório dos sujeitos na pesquisa.

Em ambas as etapas, a pesquisa foi apresentada aos desenvolvedores em uma reunião em que foram explicados os instrumentos e esclarecidas as condições de um trabalho de natureza acadêmico, com participação voluntária e confidencial.

O conjunto de FID selecionados para observação foi composto de fatores organizacionais e fatores individuais. Entre os FID organizacionais têm-se a comunicação, o relacionamento, o controle sobre os processos, as políticas de Recursos Humanos, os papéis e responsabilidades, o sistema de reconhecimento e recompensa. Entre os FID individuais, selecionaram-se: a dificuldade para o desempenho da tarefa, os problemas pessoais (financeiros, de saúde, de relacionamento), a carga de trabalho percebida, o estado emocional.

Os defeitos de software foram classificados pelos desenvolvedores dentro das seguintes opções, consideradas comuns na empresa, havendo a opção da resposta aberta para os casos não previstos: implementação incompleta dos requisitos, omissão do tratamento do requisito, problemas na integração com outros sistemas; projeto inadequado da interface; problemas na codificação/digitação e implementação inadequada da navegação.

Descreve-se a seguir, com mais detalhes, o estudo realizado.

4.1. Primeira etapa

Na primeira etapa da pesquisa, procurou-se identificar, de forma qualitativa, o conjunto de fatores influenciadores do desempenho considerados relevantes pela população de desenvolvedores e um conjunto de tipos de defeitos, que foram mapeados para situações de erro humano. Esta primeira etapa resultou em um questionário piloto submetido a um grupo de 11 desenvolvedores de software. O questionário, composto por 19 questões, caracterizava o projeto em termos gerenciais, o papel do desenvolvedor no projeto, o ânimo do desenvolvedor, sua percepção de carga de trabalho e o defeito de software encontrado. Os resultados da aplicação piloto permitiram um refinamento do instrumento de pesquisa.

Julgou-se que a visão do grupo de trabalho sobre seus próprios problemas poderia trazer o benefício de foco à pesquisa, detalhando melhor os FID reconhecidamente presentes naquela particular indústria e ignorando aqueles que fossem irrelevantes.

4.2. Segunda etapa

Baseada nos FID identificados na primeira etapa, a pesquisa de campo realizada teve por objetivo estabelecer uma base temporal destes fatores bem como analisar os defeitos de software encontrados. A intenção desta abordagem foi a de permitir que, quando um



defeito de software fosse identificado, ele fosse rastreado até sua origem, de forma que se conhecessem as condições de contexto daquela época, para o desenvolvedor responsável pelo defeito. Para tanto, desenvolveu-se um outro questionário, também disponibilizado na intranet da empresa.

O questionário foi formado por três blocos de questões. O primeiro agrupou fatores que caracterizaram a inserção do indivíduo na equipe de trabalho e na empresa, e que são, a grosso modo, independentes do projeto e do momento. As seis questões deste bloco avaliaram: a existência de um processo de software no setor, a visão do desenvolvedor sobre a cooperação nas equipes de trabalho, sua satisfação com a políticas de remuneração e recompensa da empresa e a visão sobre a capacidade do líder da equipe. Estas questões foram respondidas apenas uma vez.

O segundo bloco de questões teve por objetivo contextualizar as variáveis relacionadas às tarefas e à situação individual, que são dependentes do tempo. Este bloco do questionário foi respondido diariamente, com o objetivo de formar uma história das variáveis influenciadoras que pudesse ser resgatada quando necessário. Assim, as sete questões deste bloco contemplaram as seguintes variáveis: etapa do projeto, grau de envolvimento com o projeto, tipo das dificuldades experimentadas pelo desenvolvedor, nível de cooperação com outros elementos do grupo de projeto, existência de problemas (particulares/financeiros/de saúde), estado emocional, percepção da carga de trabalho.

O terceiro bloco teve por objetivo contextualizar a identificação e recuperação dos defeitos de software. Esta última parte também trata de variáveis dependentes de tempo e desta forma foi preenchida diariamente pelos desenvolvedores, para cada defeito encontrado. As questões caracterizaram se o defeito encontrado foi produzido pelo desenvolvedor ou por outra pessoa; em que fase do projeto ele teve origem, mais provavelmente; há quanto tempo, aproximadamente, o defeito foi produzido; a forma de detecção; o tipo do defeito; a causa, na opinião de quem o encontrou. Desta forma, fica possível comparar o defeito detectado com a situação que o gerou.

5. Resultados obtidos

Foram relatados 84 defeitos durante a vigência da pesquisa. Os resultados obtidos nesta pesquisa são qualitativos e têm apenas o objetivo de testar o protocolo experimental. Os dados colhidos não têm volume nem qualidade suficiente para um tratamento estatístico mais elaborado, por diversas razões. Em primeiro lugar, várias das situações previstas não ocorreram no período analisado. Segundo, poucos projetos foram totalmente desenvolvidos no tempo dado, o que pode levar a um viés na amostra em função das características específicas de contexto destes projetos. Finalmente, a amostra foi obtida de um grupo que não pode ser considerado representativo de toda uma população de desenvolvedores, tanto por seu tamanho quanto pelo fato de estarem todos trabalhando em uma mesma empresa. Uma amostra significativamente maior e a análise de correlação entre os fatores e os erros obtidos seriam necessárias para se estabelecer a implicação entre estes fatos. No entanto, os resultados foram considerados interessantes para a avaliação do método de pesquisa.

Em relação à inserção do indivíduo na equipe de trabalho e na empresa, observou-se que os projetos são bastante distintos em relação à maturidade dos processos de software. Existe uma forte insatisfação com a política de RH e tensões



entre os desenvolvedores e sua liderança. Embora sejam abertos à comunicação, os desenvolvedores demandam maior competência gerencial. Dentro das equipes, o relacionamento é aceitável e a comunicação é considerada boa.

Em relação às tarefas e situação individual, ao longo do tempo do estudo, observou-se que a motivação é alta. Mesmo os desenvolvedores que se dizem insatisfeitos com as políticas organizacionais têm uma visão positiva dos projetos nos quais estão inseridos. Apenas 10% consideram seu projeto entediante. A maioria dos desenvolvedores trabalha na fase de implementação.

Ao longo da pesquisa, observou-se que diversos desenvolvedores enfrentaram problemas. De forma totalizada, 54% dos relatos apontam problemas, sendo 24% de natureza profissional e 20% de natureza particular. A análise do estado emocional baseou-se no seguinte critério: foram apresentados 10 estados emocionais ao desenvolvedor, sendo que cinco deles representam estados positivos (bem, alegre, confiante, disposto, animado) e outros cinco representam estados negativos (mal, triste, nervoso, doente, desanimado). Na grande maioria dos casos o estado emocional era positivo (345 ocorrências contra 40 situações negativas).

Quanto aos defeitos encontrados no software, 71% dos defeitos foram produzidos na fase de implementação; quanto ao tipo de defeito, o mais comum nesta fase são defeitos de implementação dos requisitos, de forma incompleta ou incorreta.

Apresentam-se a seguir algumas constatações quanto aos defeitos de software e os FID. Na pesquisa realizada, a possibilidade de se tratar desenvolvedores engajados em projetos distintos trouxe alguma diversidade de situações para este trabalho, havendo desenvolvedores muito insatisfeitos, com motivação baixa ao lado de desenvolvedores com estado emocional bastante positivo e motivação elevada. Percebe-se que a comunicação, o controle sobre os processos, a satisfação, a liderança, problemas profissionais, a carga de trabalho e motivação são os fatores freqüentemente apontados como problemáticos nas situações em que há erros.

Um exemplo, apenas a título de ilustração, mostra uma desenvolvedora que durante o período analisado passou de um projeto motivante e de um estado emocional classificado em +3, para um trabalho entediante, com carga de trabalho percebida como excessiva e estado emocional definido como -2. A desenvolvedora, que não apresentava problemas, passou a se classificar como portadora de problemas profissionais. Neste período, os erros cometidos por esta desenvolvedora seguiram o gráfico apresentado na Figura 2, onde se pode observar nitidamente o crescimento do número de erros no período.



Figura 2. Erros cometidos pela desenvolvedora ao longo do estudo



Nos projetos com processo de software definido, não aconteceram defeitos de implementação redundante de requisitos e reduziram-se os defeitos de integração, em comparação com os projetos *ad-hoc*. Considerando-se que o conhecimento sobre os requisitos fique melhor num processo definido, é de se esperar que haja uma diminuição nos erros humanos de natureza informacional.

Em relação à carga de trabalho, é interessante observar que nos projetos de muito alta motivação não houve relato de carga de trabalho alta, ao contrário das demais situações. A escala Nasa-TLX [Hart, 1988] é clara em estabelecer o grau de frustração como um fator que afeta, positiva ou negativamente, a percepção de carga de trabalho. No caso, o entusiasmo age como um fator de redução da carga percebida.

Um fato esperado é que apesar de a carga de trabalho não ser considerada excessiva de forma ampla na empresa, o número de defeitos gerado mediante carga alta é proporcionalmente maior do que com carga normal. As causas de defeitos mais freqüentemente apontadas no caso de carga alta são a falta de documentação e a falta de conhecimento técnico. De fato, com carga alta, a busca pela informação necessária para cumprir a tarefa pode ser prejudicada. Os dados de carga excessiva podem estar subestimados, uma vez que quando os desenvolvedores usaram a resposta aberta para caracterizar o tipo de defeito eles registraram repetidamente o termo “falta de tempo”.

O mapeamento entre causas e conseqüências também corrobora as teorias de erro humano. Defeitos de codificação e digitação, tipicamente deslizos, são em 60% dos casos atribuídos a interrupções, assim como 41% dos problemas de implementação incompleta (tipicamente lapsos).

Já alguns fatores não mostraram qualquer relação com os erros nos dados coletados: relacionamento, papéis/responsabilidades e reconhecimento. Este fato não os invalida como FID. Dos seguintes fatores não se pode afirmar nem sua relação nem ausência de sua influência: problemas pessoais, financeiros e de saúde; estado emocional e dificuldades no desempenho de tarefas.

5. Conclusões

A intenção do estudo foi de avaliar a possibilidade de se lançar um olhar de confiabilidade humana no processo de desenvolvimento de software, visando a melhoria de sua qualidade. Este olhar implica em se considerar que o defeito existente no software tem origem num erro humano e que estes erros são sujeitos a FID que se podem observar.

Acredita-se que a monitoração da variáveis de contexto com influência no desempenho das atividades humanas de desenvolvimento de software possa levar a uma maior qualidade no produto final. Os estudos realizados levam a crer que o desenvolvimento de uma ferramenta de monitoração e classificação dos defeitos de software como erros humanos pode levar a um conhecimento de valor para a empresa, permitindo intervenções organizacionais mais acertadas. Por exemplo, ações de revisão e testes podem ser incrementadas quando os desenvolvedores sinalizarem situações de contexto sub-ótimas. Da mesma forma, esta monitoração deve permitir avaliar aqueles defeitos introduzidos no próprio processo de remoção de outros defeitos.



Mais que isso, a grande importância do entendimento do defeito de software como erro humano está em reconhecer que são necessárias investigações das causas e a remoção dos problemas comuns, por intervenções com foco no ser humano.

Referências

- Almeida Junior, J. R.; Segurança em Sistemas Críticos e em Sistemas de Informação – Um estudo Comparativo. Tese de Livre Docência. Escola Politécnica da USP, 2003.
- Dekker, S. The Field Guide to Human Error Investigations. Ashgate, 2002.
- Hart, S.G.; Staveland, L.E. Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. In: Hancock, P.A.; Meskhati, N. (Eds.) Human Mental Workload. Elsevier, 1988. p.139-183.
- Hollnagel, E. Cognitive Reliability and Error Analysis Method. Elsevier, 1998
- Hollnagel, E.; Woods, D.; Leveson, N. International Symposium on Resilience Engineering, 2004 (acessado em 09.03.2006) disponível em <http://csel.eng.ohio-state.edu/woods/error/About%20Resilience%20Engineer.pdf>
- Isaac, A.; Shorrock, S.T.; Kennedy, R.; Kirwan, B.; Andersen, H.; Bove, T. Human Error in ATM Technique – HERA JANUS. Eurocontrol, 2003.
- Nakamura, K.; Sugawara, M.; Toyama, M.; Defect prevention activities and tools Communications, ICC '91, Conference Record. IEEE International Conference on , 1991, p. 360-363
- NRC, Technical Basis and Implementation Guidelines for A Technique for Human Event Analysis (ATHEANA). Nuclear Regulatory Commission, 2000 (NUREG-1624)
- Pfleeger, S. L. Software Engineering: Teory and Practice. Prentice-Hall, 2001
- Smidts, C.; Stutzke, M.; Stoddard, R. Software Reliability Modeling: An Approach to Early Reliability Prediction. IEEE Transactions on Reliability, v. 47, p. 268-78, 1998
- Stutzke, M. A.; Smidts, C. S.; A stochastic model of fault introduction and removal during software development, IEEE Transactions on Reliability, jun 2001, p. 184-193
- Swain, A.D.; Guttman, H.E. Handbook of Human Reliability Analysis with emphasis on Nuclear Power Plant Applications. U.S. Nuclear Regulatory Commission, 1983. (NUREG/CR-1278).