

ERRO APÓS O TÉRMINO DA OPERAÇÃO NO DESENVOLVIMENTO DE SOFTWARE

Andrezza Lima ARAGÃO e Luiz Carlos BEGOSSO

andrezzaaragao@hotmail.com, begosso@gmail.com

RESUMO: Erros após o término da operação caracterizam-se por serem falhas que ocorrem durante a uma tarefa rotineira (Li, Blandford, Cairns, Young, 2005) que afetam boa parte da população pois geralmente são esquecidos pela falta de atenção dada a eles. Podem ser reconhecidos por serem erros que não necessariamente impedem que a tarefa seja concluída, mas que a mesma não seja feita com perfeito sucesso. Durante a pesquisa, foi realizado testes com alunos para simular cenários que os erros podem acontecer, 62% dos alunos que passaram por esses testes cometeram erros após o término da operação. Suspeita-se que esses erros são cometidos por falha na memória, interrupções durante a execução da tarefa, atenção e até mesmo motivação.

PALAVRAS-CHAVE:

Erros humanos; erros no desenvolvimento de software; erros de pós conclusão;

1. Introdução

Durante a construção de um software este passa por diversas etapas, que vão desde o planejamento até finalização dos últimos detalhes da integração e instalação. No entanto, mesmo com todo o empenho e preocupação na busca do software perfeito, podem

ocorrer problemas inesperados: celulares que explodem no bolso do usuário, sistemas comerciais que param e atrapalhavam a vida de muitas pessoas são exemplos em potencial de erros cometidos pelos próprios construtores.

O presente estudo que está inserido na área de erros humanos cometidos após o término de alguma operação (em inglês *post-completion error*) executada por um usuário, tem por objetivo verificar se tal tipo de erro é suscetível na área de programação de computadores. Este tipo de erro é muitas vezes ignorado, pois passa despercebido por profissionais que deveriam identificar a possibilidade de sua ocorrência.

De acordo com Li, Cox, Blandford, Cairns, Young e Abeles (2016) os erros cometidos após o término de uma operação são erros que ocorrem após a conclusão de uma tarefa principal, pois os usuários tendem a perder o foco no objetivo principal ao terem a maior parte da tarefa já concluída e acabam por anular os últimos detalhes para finalizar por definitivo a tarefa.

Um exemplo comum de erro cometido após a conclusão de uma operação, é aquele em que o usuário ao fazer cópias numa copiadora, esquece o original dentro da máquina. Ament, Lai e Cox (2011) definem esses acontecimentos como difíceis de serem eliminados por não serem erros cometidos por falta de conhecimento e sim por um lapso de esquecimento do usuário no tempo em que o procedimento é executado.

Um outro exemplo a ser citado é da área de entretenimento em que, num jogo de cartas, uma das regras é que a cada duas vezes que o jogador lançar cartas deve-se descartar uma, esquecer-se dessa condição pode ser considerado como um erro.

Na área de sistemas críticos, por exemplo, erros de pós conclusão não são admitidos, pois se um software que monitora os batimentos cardíacos de um paciente falhar tal fato pode levar a resultados trágicos.

Back, Cheng, Dann, Curzon e Blandford (2006) conduziram um estudo sobre os erros de pós conclusão e, em um dos testes aplicados na área de jogos, os autores abordaram a ideia de motivar os participantes a não cometerem os erros. Ao recolherem os resultados concluíram que mesmo com a devida motivação os participantes continuaram a cometer as mesmas falhas e as estatísticas não diminuíram sobre a quantidade de erros cometidos. Os autores concluíram que criar formas de motivação não garante que os usuários deixarão de cometer erros de pós conclusão.

Um estudo de Ament, Cox, Blandford e Brumby (2010) diz que a principal causa de erros de pós conclusão é a memória do executante da tarefa. O usuário tende a ter a

mente cheia de tarefas e acaba não se dedicando a nenhuma delas por completo. Por exemplo, enquanto ele pega uma xícara de café está pensando no trabalho que deve entregar no final do dia e acaba se esquecendo de fechar a garrafa de café. Seu objetivo principal já foi conquistado de qualquer forma, dessa forma sua mente automaticamente esquece de terminar o procedimento por completo.

2. Metodologia

No decorrer do projeto foram conduzidos dois testes com alunos do primeiro ano do curso de Análise e Desenvolvimento de Sistemas. Os alunos participantes dos testes cursavam a disciplina de Algoritmos e Estruturas de Dados I e possuíam pouca ou nenhuma experiência em programação de computadores. Os testes tiveram como intuito verificar cenários em que são mais suscetíveis a ocorrência de erros de pós conclusão.

2.1. 1º Teste

No primeiro teste 37 alunos fizeram um algoritmo como exercício que deveria seguir as regras de programação da linguagem C++. Após a realização do exercício, foram identificados os erros de pós conclusão: esquecimento de colocação de ponto e vírgula ao final de instrução e as chaves “{}” que devem ser abertas no início de uma estrutura condicional e fechada ao final dessa mesma estrutura. Caso o aluno se esqueça de colocar o ponto e vírgula ou as chaves, o programa automaticamente não é compilado mesmo que a lógica do exercício esteja correta. Geralmente os alunos tendem a se preocuparem mais com a lógica do algoritmo, ou seja, a meta principal sem prestar muita atenção nos pequenos detalhes que devem executar para então concluir a tarefa com sucesso. Por conta disso, declarar como erro de pós conclusão a falta dos pontos e vírgulas e chaves se torna oportuna para a ocasião, pois os humanos estão sempre tão focados na grande missão que seu cérebro acaba automaticamente por descartar o pensamento que um “ponto e vírgula” ou uma chave “ {} ” tenha a chance de causar um erro em todo o projeto. A Tabela 1 apresenta uma compilação dos resultados do 1º Teste.

Tabela 1 - Resultados do 1º Teste

Tipos de Situações	Cometeram Erros	Não cometeram	Erros cometidos mais de uma vez
Erro de ponto e virgula	40,5%	59,5%	16,2%
Erros de falta de chave	62,2%	37,8%	13,5%

A partir dos dados da Tabela 1, observa-se que o erro de pós conclusão com maior ocorrência está no fato do esquecimento de abrir ou fechar chaves. Tal fato pode ser explicado da seguinte forma: os estudantes estão habituados a escrever programas utilizando um editor de texto para a linguagem C++, e este editor automaticamente fecha chave a partir de um comando de abertura de chave. Assim, pode-se concluir que o fato de estarem habituados a usar o programa pode ter causado alguma dependência.

Borges e Perini (2003) destacam que a falta de conhecimento diante a uma atividade ou tarefa pode conduzir a erros de pós conclusão. Essa constatação é verdadeira a medida em que sabe-se da pouca ou nenhuma experiência dos alunos com a área de programação de computadores.

2.2. 2º Teste

No 2º Teste também foi realizado um exercício voltado ao ambiente de programação do C++. Escolheu-se aleatoriamente 46 alunos da mesma série e curso do 1º Teste e estes realizaram as tarefas algumas semanas depois da execução do 1º Teste. O intuito ainda era verificar se eles esqueceriam os pontos e vírgulas e as chaves “{}” durante a construção do algoritmo. Segundo Matavelli e Cardoso (2010) é possível afirmar que quanto menos tempo um profissional está executando sua área de serviço, mais ele tem chances de cometer erros, seguindo essa linha de raciocínio esperava-se que com o intervalo de tempo dado entre o 1º e o 2º teste os alunos estariam mais habituados com a programação e assim cometeriam menos erros. A Tabela 2 apresenta uma compilação dos resultados do 2º Teste.

Tabela 2 – Resultados do 2º Teste

Situações	Cometeram Erros	Não cometeram	Erros cometidos mais de uma vez
Erro de ponto e vírgula	36,9%	63,1%	13%
Erros de falta de chave	45,6%	54,4%	23,9%

Com os resultados da Tabela 2 nota-se que a oportunidade de os alunos ganharem mais conhecimento na área de programação durante o tempo que se seguiu até o 2º Teste tenha os ajudado a cometerem menos erros durante a realização dos algoritmos, mas que mesmo dessa forma apenas o conhecimento aprofundado sobre o assunto não tenha levado a solução absoluta dos erros. Sendo assim, é possível afirmar que apenas o aprendizado avançado sobre o assunto tratado nas execuções das tarefas não é uma solução que abrange todas as outras situações onde o erro de pós conclusão pode ocorrer. A partir dessa constatação e em consonância com os estudos de Byrne & Bovair, (1997); Byrne & Davis, (2006); Chung & Byrne, (2008) pode-se verificar que os alunos que cometeram erros de pós conclusão tendem a esquecer de executar o passo final da tarefa depois que a missão principal já foi realizada.

3. Discussão e Conclusão

Por meio dos testes realizados durante o projeto foi possível concluir que os estudantes, alvos dessa pesquisa, cometeram diversos deslizes o que acabou resultando em erros de pós conclusão. Nos casos de erros por falta de ponto e vírgulas no primeiro teste, houve 40,5% de casos cometidos e uma queda de 0,6% no segundo teste o que leva-se a concluir que mesmo com o tempo ocorrido durante o primeiro teste e o segundo a prática dos exercícios não foi suficiente para a omissão dos erros.

Norman (1981) identifica que essas falhas podem ser explicadas por conta de uma incapacidade do usuário em realizar um monitoramento da atenção necessária para cada passo da tarefa. No caso dos erros de falta de chave, no experimento aqui conduzido, destaca-se que houve uma redução de 16,6% se comparado aos erros de esquecimento de

ponto e vírgula. Tal situação demonstra que mesmo não sanando os erros cometidos, existem formas de diminuir sua frequência.

Segundo a visão de Huang (2016, p. 02) “Apesar de ser poucos os estudos sobre erro pós conclusão em ciência da computação, é recomendável focar os usuários a melhores regras de design de interfaces homem-computador”. Na criação de novas máquinas, seria recomendável deixar com que cada tarefa que o usuário possa fazer na utilização do aparelho possa ser evidenciada. Por exemplo, a máquina de autoatendimento de um banco, ao colocar o cartão de crédito uma luz se acende, no entanto, seria útil se fosse criado um sistema que ao notar que o usuário está finalizando suas operações emitisse um sinal sonoro para que ele não se esqueça do cartão ainda inserido no aparelho.

Erros de pós conclusão fazem parte de uma abrangente lista de causas como foi retratado acima e sem muitas soluções definitivas para as situações que eles se apresentam, no entanto, na área de desenvolvimento de software já existe uma ferramenta que pode contribuir com a diminuição da constância dos acontecimentos. Conhecido como Poka Yoke – à prova de erros – é utilizado para reduzir as chances de erros no processo produtivo, sejam estes gerados por pessoas ou máquinas (PANGARTTE, 2014). Os sistemas Poka Yoke são utilizados para corrigir ou prevenir erros e para isso existe um método para cada tipo de erro: Métodos de Advertência que são conhecidos como os que não param a execução da tarefa, mas emitem um sinal como luzes ou alarmes demonstrando que há um erro e Métodos de Controle que para todo o processo em execução impedindo que a máquina vá para a etapa seguinte caso exista um erro. Segundo Pangartte (2014) após os testes feitos em seu projeto utilizando os sistemas Poka Yoke os resultados demonstraram que o sistema pode trazer ganhos significativos em diversos aspectos.

As razões que levam esses erros acontecerem são variadas, John e Kieras (1996, p. 02) dizem que “nenhuma metodologia desenvolvida de prevenção aos erros foi afirmada como suficiente para a resolução do problema”. Por ser um erro constante que pode ser gerado por um leque de situações simples como passar dados para um pen drive e esquecer o dispositivo plugado no notebook, existe uma dificuldade de encontrar uma solução universal. Conclui-se que erros de pós conclusão estão cotidianamente na vida dos humanos e sendo assim das máquinas e uma solução para estas situações ainda não foi encontrada, no entanto, é possível assegurar que enquanto essa solução não é descoberta há meios de prevenir que os erros de pós conclusão ocorram.

4. Referências Bibliográficas

AMENT, Maartje G. A., COX, Anna L., BLANDFORD, Ann, BRUMBY, Duncan. Working Memory Load Affects Device-Specific but Not Task-Specific Error Rates. COGNITION IN FLUX, pg. 91-96, Portland, 2010.

_____AMENT, Maartje G. A., LAI, Anita Y. T., COX, Anna L. The Effect of Repeated cue exposure on Post-Completion errors. CogSci 2011, pg 850-855, Boston Massachusetts.

BACK, Jonathan, CHENG, Wai Lok, DANN, Rob, CURZON, Paul, BLANDFORD, Ann. Does being motivated to avoid procedural errors influence their systematicity? HCI 2006 Vol.1.

BORGES, M., PERINI, E. (2003). E Medication errors: Who is responsible?, Revista da Associação Médica Brasileira. V.49, (3), 335-341.

BYRNE, Michel D., BOVAIR, Susan. A Working Memory Model of a Common Procedural Error. Cognitive Science, Pittsburgh, 1997, pg. 31-61.

_____BYRNE, M. D., DAVIS, E. M. (2006). Task structure and postcompletion error in the execution of a routine procedure. Human Factors, 48 (4), pg. 627–638.

CHUNG, Phillip H., BYRNE, Michael D. Cue effectiveness in mitigating postcompletion errors in a routine procedural task. Journal International Journal of Human-Computer Studies, Duluth, pg. 217-232, 4 Abril 2008.

HUANG, Fuqun. Post-completion Error in Software Development. 9th International Workshop on Cooperative and Human Aspects of Software Engineering, At Austin, Texas, 2016.

John, B. E., & Kieras, D. E. (1996). Using GOMS for user interface design and evaluation: Which technique? ACM Transactions on Computer-Human Interaction, 3 (4), 287–319.

LI, Simon Y. W., BLANDFORD, Ann, CAIRNS, Paul, YOUNG, Richard M. Post-completion errors in problem solving. XXVII annual conference of the cognitive science society July 21-23 Stresa, Italy, 2005.

_____LI, Simon Y. W., COX, Anna L., BLANDFORD, Anna L., CAIRNS, Paul, YOUNG, Richard M., ABELES, Aliza. Further investigations into post-completion error:

the effects of interruption position and duration. Proceedings of the Annual Meeting of the Cognitive Science Society, 33, Londres, 2016. Disponível em: <<http://escholarship.org/uc/item/03d43898>> Acesso em: 27/07/2017.

MATAVELLI, Rafaela Dias, CARDOSO, Ana R. O Erro humano no contexto da Saúde: A antiguidade na organização e a percepção do erro nos Enfermeiros, 2010. 8º Congresso Nacional de Psicologia da Saúde, At Lisboa.

NORMAN, D. A. Categorization of action slips. Psychological Review, Vol. 88 (1), pg. 1-15, 1981, University of California, San Diego (La Jolla).

PANGARTTE, Reginaldo. Sistemas Poka Yoke para processos de teste em Indústria Eletroeletrônica, 2014. Universidade Tecnológica Federal do Paraná Departamento Acadêmico de Mecânica.