



**UNIVERSIDADE FEDERAL
DE SANTA CATARINA**

O Desenvolvimento da Tecnologia; A Obsolescência dos Softwares CAD

Leandro Neves – leandro.nve@gmail.com

Victor Hass Severo – victorhass@gmail.com

Resumo:

Na engenharia, a grande maioria dos projetos requerem o uso de softwares CAD para fins de simulação e prototipagem. No entanto, com o desenvolvimento destes softwares, a necessidade de atualizações levanta uma séria questão do contexto atual da sociedade: seriam todas estas atualizações realmente necessárias e aprimoram a funcionalidade destes softwares ou algumas destas atualizações na verdade prejudicam a interoperabilidade dos programas?

1. Introdução

O conceito de obsolescência visa abranger um aspecto amplo e geral da perda total ou da diminuição da capacidade de ser útil, sobretudo em softwares do tipo CAD (Desenho Assistido por Computador), nos quais a interatividade com diversos terminais e possíveis erros de parâmetros podem ser vistos como cruciais para um projetista. A obsolescência do software pode gerar entropia suficiente para desestruturar um projeto de anos.

O software do tipo CAD é um programa paramétrico que consegue definir, a partir de dados sobrepostos (de entrada), uma visualização virtual de um esboço 2D ou geometria 3D¹. O mesmo, para ser executado, exige determinados requisitos de sistema.

Isto pode ser comparado com casos mais palpáveis e impactantes na sociedade contemporânea, principalmente em se tratando do conceito de obsolescência planejada, em que a procura pelo lucro acaba levando os fabricantes a projetar de modo que o produto não seja mais útil após uma determinada faixa de tempo. Analogamente, os problemas de compatibilidade entre versões distintas dos softwares CAD podem muito bem levar os consumidores a sempre buscar a versão mais recente destes produtos, o que cairia “como uma luva” para os desenvolvedores de tais softwares.

¹ Duas e três dimensões respectivamente.

2. Problemática

O principal aspecto da problemática decorre da falta de utilidade de um software CAD devido a uma atualização, isto é, a obsolescência do mesmo, característica observada quando a atualização do programa entra em conflito com fatores físicos, virtuais e econômicos. É possível, ainda na mesma vertente, compreender a necessidade de um melhoramento do programa na problemática, porém isto não deve afetar sua utilidade perante a confecção de projetos de engenharia.

Visto sob um panorama do desenvolvimento tecnológico, Pfleeger (2001) demonstra que o foco da intenção em uma atualização está em produzir algo que atenda aos requisitos e funcione corretamente. Mas como determinar parâmetros de atualização para todos os usuários? O gerenciamento de uma atualização global captaria também os usuários avançados²?

Segundo Nunes (2005), os problemas desenvolvidos com atualizações frequentes de softwares é o da capacidade de gerar ainda mais problemas, assim Nunes visa esclarecer que uma atualização, por conseguir agregar mais informações a um sistema com tantas ramificações, têm igual capacidade e responsabilidade na geração de mais problemas advindos do mesmo.

² Que fazem suas próprias modificações e aprimoramentos.

2.1. Necessidade de uma atualização?

Sem a imposição (ou inteira responsabilidade) de recorrer ao consumo compulsivo³ para conseguir explicar o fenômeno do surgimento de novas atualizações de CAD, onde os mesmos se justificam pela prerrogativa de corrigir diversos tipos de problemas, causados ou não por si próprios, e *bugs* que surgem na utilização de uma versão do software, podendo realizar a inclusão de novas funcionalidades, novos recursos, *patches*⁴ de linguagem, entre outros. A atualização pode ser vista como uma manutenção em uma máquina, divergente apenas natureza da manutenção a ser realizada.

Kessler e Mendes (2015) alertam para os riscos de não se atualizar softwares que, entre diversos fatores, podem prejudicar a comunicação interna com o hardware, causar a perda de funcionalidades, além de comprometer significativamente a segurança digital do terminal.

As atualizações de software desempenham diferentes funções. Elas estão disponíveis para o sistema operacional e para programas de software individuais. A realização dessas atualizações implantará várias revisões em seu computador, como a adição de novos recursos, a remoção de recursos desatualizados, a atualização de drivers, a distribuição de correções para bugs e, o mais importante, a correção das brechas de segurança recém-descobertas (Norton).

Entretanto, sendo mais específico para o segmento CAD: ao mesmo tempo que uma atualização pode trazer ao software melhora na interoperabilidade com

³ Segundo Marx, o valor de uso está relacionado à utilidade da mercadoria. Ou seja, o valor de uso é diretamente a base material onde se apresenta uma relação econômica determinada: o valor de troca. E a base do valor de troca, ou do valor propriamente dito, é o trabalho humano necessário para produzir essas mercadorias. Marx mostra que, de forma alguma, ocorre uma troca equivalente no processo de compra e venda da força de trabalho: o que ocorre é o fetiche da mercadoria, de modo que o produto direto do trabalho (sob a forma de mercadoria), o próprio capital e as relações sociais que ele engendra, "parecem dotados de vida própria, figuras autônomas, que mantêm relações entre si e com os homens" e que fazem as relações próprias do capitalismo. parecer como naturais e eternas. MARX, Karl. O Capital - Volume I Livro I. (Os Economistas) São Paulo: Editora Nova Cultura, 1996

⁴ Um *patch* (termo da língua inglesa que significa, literalmente, "remendo") é um programa de computador criado para atualizar ou corrigir um *software*. Esta técnica pode ser utilizada como uma das formas de distribuir uma liberação de *software*.

softwares similares, ele realiza o efeito inverso com si próprio. O que ocorre é que os novos arquivos gerados não podem ser executados nas versões antigas, e, uma vez que um arquivo de uma versão desatualizada é 'convertido' para a mais recente, ele se torna inexecutável na versão antiga. O problema é ainda maior quando visto sob um trabalho em equipe: todos os terminais precisam ser atualizados para que nada seja perdido e obtenham uma perfeita integração.

É observado que tecnologias revolucionárias sofrem certa resistência em sua inserção no meio social, o que ocasiona um longo período de adaptação. (Luca e Balconi, 2016)

Uma outra forma de enxergar o interesse na atualização é sob uma ótica econômica: é fato que uma desenvolvedora necessita captar recursos. Atualizações de softwares conferem à desenvolvedora o poder de tornar a *bel prazer* um produto parcialmente ou inteiramente obsoleto.

2.2. Relação Usuário-Desenvolvedor

Sob a perspectiva de mercado, as desenvolvedoras de software precisam continuar lucrando, e, uma vez que exista a venda de um produto, é possível que coexistam determinados tipos de relações:

- A comercialização em conjunto com atualizações de manutenção periódicas e gratuitas (por uma determinada faixa de tempo, por exemplo, até que existam pessoas trabalhando em outros recursos pagos para o software);
- A venda somente do software 'base', ou seja, com a venda separada de recursos que aprimoram o desempenho de utilização;
- Ou ainda, as duas opções supracitadas juntas.

Os usuários que adquirem atualizações estão incentivando indiretamente o desenvolvimento tecnológico dos softwares, e ainda podem desfrutar de uma aplicabilidade direta desta tecnologia.

Quando uma atualização ou uma nova versão do software é tão indispensável a ponto de reduzir a utilidade para com a necessidade primordial de se obter o software em um dado momento, evidencia-se então o avanço tecnológico do segmento.

2.3. Como determinar que um software CAD é enfim obsoleto?

A perda de utilidade de um do software CAD é uma questão diretamente relacionada com o desenvolvimento tecnológico do mesmo, mas uma vez que possam surgir discordâncias com a culminância da interoperabilidade parcial ou completa ou ainda uma discordância com o hardware, entre diversas formas de abordar o problema, é notório que o software ficará ultrapassado.

Entretanto, é preciso deixar claro que os requisitos do sistema não conseguem determinar com supremacia se um software CAD poderá ser operado livre de defeitos e só constituem, mais uma vez, um parâmetro de comparação com o que se espera de um sistema de hardware para determinado CAD. Comprovando assim, ainda na mesma linha de raciocínio, que os requisitos do sistema também podem inviabilizar uma atualização.

Um fator importante relacionado é a falta de previsão na atualização de um software. Este fato está relacionado à influência que o software sofre de fatores externos, naturalmente imprevisíveis (Bhatt et al., 2004).

Além disso, não só para expandir mercado, mas também para suprir a queda na venda de seus produtos, o desenvolvimento de novos recursos e funcionalidades para os softwares torna-se uma atualização indispensável. Existe, então, uma restauração, ou melhor dizendo, uma manutenção da relação entre desenvolvedores e usuários, baseada unicamente no mutualismo de mercado e de quebra, desempenhando avanço tecnológico.

3. Conclusões

Fatores que geram a obsolescência:

- Interoperabilidade;
- Requisitos mínimos incompatíveis.
- Atualizações compulsórias.

A gama de dispositivos e hardwares com os quais os programas CAD tem que se comunicar só aumenta, o que leva a uma alta demanda de atualizações de compatibilidade a fim de integrar estas novas configurações, porém isto muitas vezes leva à quebra de compatibilidade de hardwares antigos devido à elevação dos requisitos mínimos dos programas.

O segmento já consolidado agora aposta em vertentes mais específicas do mercado, e a partição de nichos menores e melhor direcionados conseguiria contemplar uma maior captação monetária, ou seja, os programas CAD devem focar em adaptar-se ao novo modelo da década atual, em especial quando falamos da integração do usuário com os dispositivos móveis (principalmente smartphones), sendo necessário, assim, uma adaptação dos softwares para que a integração dos novos hardwares portáteis não leve à inoperabilidade de hardwares mais “antigos”, o que se pode notar, por exemplo, em *apps* como o AutoCAD 360, que integra pela “nuvem” as versões desktop e mobile do seu programa, permitindo assim melhor interação entre o usuário e o projeto a ser desenvolvido.

4. Referências

BHATT, P.; SHROFF, G.; MISRA, A. K. (2004) “Dynamics of software maintenance”, ACM SIGSOFT Software Engineering Notes, v. 29, n. 5, p. 1-5.

KESSLER, Luiz Felipe; MENDES, Eduardo. Dicas para proteger seu PC de softwares indesejados. 2015. Disponível em:

<<https://www.oficinadanet.com.br/post/13901-dicas-para-proteger-seu-pc-de-softwares-indesejados>>. Acesso em: 15 nov. 2016.

LUCA, Artur Back de; BALCONI, Téo Bernal. Desenvolvimento Tecnológico e Relações de Trabalho. NEPET: Tecnologia e Desenvolvimento. Florianópolis - Sc, p. 11. 24 jun. 2016.

MARX, Karl. O Capital - Volume I Livro I. (Os Economistas) São Paulo: Editora Nova Cultura, 1996

NORTON, by Symantec. A importância das atualizações e patches de software. Disponível em: <<https://br.norton.com/importancia-das-atualizacoes/article>>. Acesso em: 15 nov. 2016.

NUNES, J. D. (2005) “Projetos de Planos Pedagógicos Orientados a Problemas”, Biblioteca Digital da SBC. Disponível em <<<http://www.sbc.org.br/bibliotecadigital/download.php?paper=218>>>. Acesso em: 16 nov. 2016.

PLEEGER, S. L. (2001) “Software Engineering: theory and practice”. Second Edition, New Jersey, Prentice Hall.