ESCOLA DE APERFEIÇOAMENTO DE OFICIAIS DA AERONÁUTICA
COORDENADORIA ACADÊMICA
CURSO DE APERFEIÇOAMENTO DE OFICIAIS 1º/2020

ISRAEL CORDEIRO DOS SANTOS ROCHA, Cap Eng

OBSOLESCÊNCIA DE SOFTWARE:
estágio essencial no ciclo de vida dos sistemas da Força Aérea

Rio de Janeiro
2020

ESCOLA DE APERFEIÇOAMENTO DE OFICIAIS DA AERONÁUTICA
COORDENADORIA ACADÊMICA
CURSO DE APERFEIÇOAMENTO DE OFICIAIS 1º/2020

ISRAEL CORDEIRO DOS SANTOS ROCHA, Cap Eng

OBSOLESCÊNCIA DE SOFTWARE:
estágio essencial no ciclo de vida dos sistemas da Força Aérea

Trabalho de conclusão de curso apresentado no Curso de Aperfeiçoamento de Oficiais da Aeronáutica como requisito parcial para aprovação no Curso de Pós-graduação em Gestão Pública com ênfase em Gestão de Projetos e Processos. Linha de Pesquisa: Administração Militar. Orientador: Maj. Int. Rogério dos Santos Ferreira

Rio de Janeiro
2020

ISRAEL CORDEIRO DOS SANTOS ROCHA, Cap Eng

OBSOLESCÊNCIA DE SOFTWARE:
estágio essencial no ciclo de vida dos sistemas da Força Aérea

Trabalho de conclusão de curso apresentado no Curso de Aperfeiçoamento de Oficiais da Aeronáutica.

Aprovado por:

_____

Rogério dos Santos Ferreira, Maj Int
EAOAR

_____

Thiago Diorgilis Ribeiro Daniel, Maj Av
EAOAR

_____

Daniel Rodrigues Figueiredo, Maj Av
EAOAR

Rio de Janeiro
Julho de 2020

RESUMO

Cada aplicativo ou sistema informatizado (*software*) utilizado na Força Aérea Brasileira (FAB) deve ter sua obsolescência identificada e priorizada em uma fila de manutenção evolutiva, de substituição ou até mesmo de desativação. Tal ação reduzirá riscos de potenciais danos causados por ataques cibernéticos e de inadequação face a mudanças nos dispositivos normativos e legais que embasam a regra de negócio desses aplicativos. A obsolescência de um *software* se inicia ao término de seu desenvolvimento quando se estagna sua evolução. Assim, os elos de Tecnologia da Informação (TI) devem assessorar e sensibilizar suas organizações militares a observarem a descontinuidade e falhas de segurança em cada componente de tecnologia utilizado em seus *softwares*. Devem ainda monitorar a evolução dos dispositivos normativos e legais que regem suas regras de negócio e informar à Diretoria de Tecnologia da Informação (DTI) sempre que alguma obsolescência for detectada, para que, de forma concisa e coordenada, sejam priorizados os esforços de manutenção, substituição ou desativação. Este ensaio é de parecer que seja feito o mapeamento das tecnologias e dispositivos normativos e legais de todos os *softwares* utilizados na FAB, de forma gradual, seguido de uma análise criteriosa de obsolescência e de uma priorização em uma fila de manutenção de todos aqueles que necessitarem ser atualizados, de forma a garantir conformidade normativa e segurança. É essencial que tal análise seja atualizada durante todo o ciclo de vida dos sistemas para garantir a soberania digital da Força Aérea Brasileira.

Palavras-chave: Falha de segurança. Desenvolvimento. Manutenção evolutiva. *Software*.

1 INTRODUÇÃO

O ciclo de vida de um *software* precisa ser gerenciado desde a concepção de sua necessidade operacional (NOP) até sua desativação ou substituição (BRASIL, 2007). Durante sua operação, é essencial identificar a obsolescência desse *software* e mitigá-la, quando ocorrer, através de uma fila priorizada de manutenção evolutiva ou substitutiva, de modo que o aplicativo possa continuar operando de forma segura, estável e aderente aos requisitos normativos e legais, garantindo assim, a soberania digital da Força Aérea Brasileira (FAB).

Um *software* se torna obsoleto quando não mais atende à necessidade do cliente, seja do ponto de vista do negócio (DEVERAUX, 2010), de segurança ou de mercado (RAJAGOPALA; ERKOYUNCUA; ROYA, 2014). Observa-se, por experiência, que tal obsolescência pode ser causada por vários fatores, dos quais preponderam a descontinuidade de tecnologias nas quais o *software* se baseia e a mudança de normas e legislações. Quanto à tecnologia, as versões do sistema operacional, da linguagem em que o *software* foi escrito ou as bibliotecas e frameworks que utiliza podem ser descontinuadas e assim, atualizações de segurança (*patches*) não mais serão disponibilizadas pelos fabricantes. Ademais, os dispositivos normativos e legais que regem suas regras de negócio também podem sofrer alterações tornando o aplicativo obsoleto “da noite para o dia”.

Este ensaio aborda as principais causas de obsolescência de *software*, que se inicia com o término de seu desenvolvimento, quando é interrompida sua evolução; e ressalta que essa obsolescência deve ser identificada e priorizada no âmbito da FAB para que cada eventual evolução ou manutenção seja planejada de acordo com uma fila e com os objetivos de reduzir os riscos de potenciais danos causados por ataques cibernéticos e de readequar o *software* a mudanças nos dispositivos normativos e legais.

2 DESENVOLVIMENTO

2.1 Obsolescência Tecnológica

Segundo Rajagopala; Erkoyuncua; Roya (2014, p. 76), *softwares* são programas, procedimentos, regras, dados e documentação que funcionam em conjunto com sistemas de *hardware* e infraestrutura. Eles estão presentes em projetos de defesa, sistemas de comando, controle e comunicação, de planejamento da missão, aplicativos de gestão documental, armas complexas como um caça de combate, dentre outros. Um *software* apresenta diferentes níveis de complexidade e interdependência dos seus componentes e, segundo os autores, sua obsolescência ocorre quando alguma de suas partes deixa de ter suportes regulares de atualizações e correções (*patches*), comprometendo sua segurança (*security*) e aumentando o risco de potenciais danos causados por ataques cibernéticos.

Versões de componentes de *software* que não mais recebem *patches* de segurança e com falhas conhecidas constituem pontos de vulnerabilidade a ataques cibernéticos que podem gerar prejuízos financeiros e estratégicos incalculáveis (RUAS, 2017). Imagine os transtornos e prejuízos que causaria uma invasão ao Sistema Informatizado de Gestão Arquivística de Documentos da Aeronáutica (SIGADAER) da Diretoria de Administração do Pessoal (DIRAP) ou do Gabinete do Comandante da Aeronáutica (GABAER). Imagine se as horas de voo de todas as aeronaves de uma Ala, registradas no sistema ÓPERA, fossem alteradas. Ou ainda se uma falha em um simples componente do Portal de Autenticação de Documentos (ADOC) abrisse uma porta dos fundos (*backdoor*) para uma invasão do servidor e a partir dele de toda a INTRAER. Isso é possível e quanto mais elos fracos existirem nessa cadeia de componentes, maior a probabilidade de um ataque ser efetivo (RUAS, 2017).

Em 2013, o governo norte-americano solicitou que internautas desabilitassem o *plugin* Java em seus navegadores de Internet, em razão de uma grave falha que acometia a versão 7 da linguagem (ORACLE, 2013). Isso fora as frequentes vulnerabilidades encontradas no sistema operacional Windows da Microsoft (HÁRAN, 2020), que requerem que os *patches* lançados sejam

prontamente aplicados. No caso dos sistemas operacionais, dificilmente esses *patches* afetam a usabilidade e estabilidade do *software* e podem ser aplicados sem necessidade de modificar seu código-fonte. No entanto, se houver falha em uma biblioteca ou *framework* utilizado pelo *software*, o código-fonte precisará ser alterado para a versão corrigida do componente para então ser recompilado, testado, homologado e reinstalado.

Na Força Aérea Brasileira, há sistemas estagnados em versões obsoletas de navegadores de Internet, como, por exemplo, certos módulos do Sistema Integrado de Logística de Materiais e de Serviços (SILOMS), que dependem de tecnologias descontinuadas como o Oracle Forms 11g (OLIVEIRA, P. R. M. 2016), Adobe *Flash*, Java *Applets* e o *Netscape Plugin Application Programming Interface* (NPAPI), cujo suporte foi removido nas versões atuais dos navegadores (WILLIANS, 2017).

Podemos citar também, o SIGADAER, que passou por várias atualizações para manter-se tecnologicamente atualizado, estando na versão 8 da linguagem de programação Java, a qual ainda recebe correções de segurança e segue isenta de falhas graves como a que acometeu sua sétima versão. Esforços de atualização tecnológica como esse são necessários para evitar a obsolescência, pois, independente se ela afeta diretamente o código-fonte ou não, se os componentes, tanto de *software* quanto de *hardware*, não forem atualizados para versões mais recentes, chegará o dia em que não mais haverá suporte e *patches* de correção e assim, novas vulnerabilidades descobertas continuarão lá para serem exploradas, podendo acarretar “prejuízos incalculáveis” (RUAS, 2017).

Além da obsolescência tecnológica, Rajagopala; Erkoyuncua; Roya (2014, p. 76) identificaram dois outros tipos: funcional e logística. No primeiro, alterações no *hardware*, em algum componente ou em outro *software* no mesmo sistema, podem comprometer a segurança em um efeito em cascata. O segundo tipo ocorre, por exemplo, quando algum componente físico (*hardware*) deixar de suportar o *software*.

Em todos esses casos, o *software* deixará de ser aderente à necessidade operacional da FAB, na medida em que tornar-se-á potencialmente parte do

problema e não da solução. Assim este ensaio é de parecer que os elos de TI devem monitorar cada componente, seja de *software* e *hardware*, utilizados nos sistemas da força aérea, avaliando a descontinuidade de suas versões e descobertas de falhas de segurança. Sempre que uma falha grave for identificada, a Diretoria de Tecnologia da Informação (DTI), órgão do Sistema de Tecnologia da Informação (STI) subordinada ao Comando-Geral de Apoio (COMGAP), deverá ser notificada para que o *software* afetado seja avaliado e coerentemente inserido em uma fila de manutenção corretiva, tendo sua prioridade determinada com base na severidade da falha e probabilidade de ser explorada (DRAPER, 2019).

2.2 Obsolescência de Negócio

A obsolescência também pode ser causada por mudanças nos dispositivos normativos e legais. De acordo com Devereaux (2010, p. 55), quando isto ocorre é preciso um projeto de mitigação que começa por identificar a versão de cada norma utilizada na especificação do *software* — o que pode ser denominado rastreabilidade dos requisitos de negócio. Após isto, é preciso descobrir as diferenças entre a versão utilizada nos requisitos de negócio e a atual para então categorizá-las, indicando se elas devem ser feitas, se não podem ser realizadas ou ainda quão difícil será a mudança (DEVERAUX, 2010).

Um exemplo dessa obsolescência ocorreu nas Olimpíadas em 2016 no Rio de Janeiro, quando um decreto presidencial mudou os valores das diárias civil e militar pagas a servidores em missão nessa localidade (LEAL, 2016). Razão pela qual o sistema Onix, que se encontra em processo de substituição pelo Sistema de Concessão de Diárias e Passagens (SCDP), precisou ser emergencialmente corrigido para se adequar ao novo dispositivo legal. Quando o decreto foi assinado e publicado em 2016, o Onix se tornou “obsoleto da noite para o dia” e assim permaneceu até ser atualizado. Assim também foi com o SIGADAER em 2019, quando as espécies documentais previstas na ICA 10-1 (BRASIL, 2010) foram alteradas e até mesmo excluídas pela NSCA 10-2 (BRASIL, 2019). Avalie o transtorno que seria uma diária ser incorretamente calculada ou um processo de reserva ser rejeitado pela DIRAP por conter uma espécie documental não mais prevista.

Mudanças que afetam o negócio são comuns e devem ser aceitas com naturalidade e tratadas prontamente (OLIVEIRA, W. 2016). Ao desenvolver um *software*, deve-se, em sua gestão de riscos, mapear todas as legislações que lhe sejam afetas (DEVERAUX, 2010) e avaliar a probabilidade e criticidade de mudança. Observa-se, pela experiência, que mudanças normativas geralmente são passíveis de ser percebidas antes de entrarem em vigor e assim, este ensaio recomenda que toda possibilidade de alterações afetas ao negócio seja informada à DTI assim que identificada, para avaliação de riscos e priorização de eventuais esforços visando a manutenção corretiva ou a substituição do *software* afetado. A FAB precisa estar preparada para, de forma proativa, priorizar a fila de manutenção considerando também a obsolescência de negócio.

O presente ensaio é de parecer que é essencial identificar as obsolescências tecnológicas e de negócio nos sistemas da FAB; e priorizar de forma planejada e coerente cada manutenção, a qual deve preceder, em importância, o desenvolvimento de novos *softwares* e funcionalidades (DEVERAUX, 2010), uma vez que afetam a operacionalidade, estabilidade ou segurança de sistemas já inseridos na rotina e nos processos das organizações militares.

3 CONSIDERAÇÕES FINAIS

A obsolescência de *software* começa quando termina seu desenvolvimento. Dois fatores são preponderantes para isto: depreciação tecnológica e mudanças nos dispositivos legais. No primeiro, componentes de tecnologia utilizados na construção do *software* são descontinuados ou ainda uma vulnerabilidade grave é descoberta. No segundo, mudanças nos dispositivos normativos e legais tornam o *software* incompatível com a realidade imposta pelas novas regras de negócio.

Deixar o *software* “envelhecer” sem um planejamento de manutenção é certeza de que um dia algum componente deixará de ter suporte e *patch* de correção e assim, uma vulnerabilidade que venha a ser encontrada, não será corrigida e permanecerá como ponto de possível ataque, o qual poderá comprometer não só as informações ali contidas, como até mesmo toda a rede, causando prejuízos financeiros e estratégicos incalculáveis.

Da mesma forma, mudanças nos dispositivos normativos e legais podem tornar um *software* obsoleto “da noite para o dia” causando prejuízos substanciais aos usuários e clientes que vão desde um cômputo errado de diárias até uma espécie documental constante em um processo ser recusada por não mais existir.

Este ensaio apresentou os fatores preponderantes para a obsolescência de *software* e os decorrentes riscos de ataques cibernéticos e de inadequação ao negócio face a mudanças nos dispositivos normativos e legais. Considera, o presente trabalho, ser imprescindível que a obsolescência de *softwares* seja identificada e priorizada no âmbito da Força Aérea Brasileira, tanto a tecnológica quanto a de negócio, para que o STI avalie os impactos e priorize os esforços de manutenção em uma fila coerente que traga o *software* às condições corretas e seguras de uso. Recomenda que tal manutenção tenha precedência em relação ao desenvolvimento de novos *softwares*, bem como de novas funcionalidades, uma vez que a obsolescência afeta sistemas já inseridos nas rotinas e processos das organizações militares. Tal postura reduzirá o risco de potenciais danos causados por ataques cibernéticos a componentes vulneráveis do *software*. Ademais, a proativa identificação e priorização da obsolescência permitirá adequação mais eficiente do *software* a mudanças nos dispositivos legais, trazendo-o de volta à operacionalidade.

Para isso, o presente ensaio é de parecer que seja procedido o mapeamento das tecnologias e dispositivos normativos e legais de todos os *softwares* utilizados na FAB, de forma gradual, seguido de uma análise criteriosa de obsolescência e de uma priorização em uma fila de manutenção de todos aqueles que necessitarem ser atualizados, de forma a garantir conformidade normativa e segurança a nossos sistemas. Tal análise deve ser atualizada durante todo o ciclo de vida dos sistemas e é essencial para garantia da soberania digital da Força Aérea Brasileira.

REFERÊNCIAS

BRASIL. Ministério da Defesa. Comando da Aeronáutica. Portaria nº 129/GC4, de 5 de março de 2007. Aprova a Diretriz que dispõe sobre Ciclo de Vida de Sistemas e Materiais da Aeronáutica (DCA 400-6). **Diário Oficial da União**, Brasília: seção 1, Brasília, DF, p. 24, 07 mar. 2007.

BRASIL. Ministério da Defesa. Comando da Aeronáutica. Portaria nº 89/5EM, de 7 de junho de 2010. Aprova a Instrução que dispõe sobre Correspondência e Atos Oficiais do Comando da Aeronáutica (ICA 10-1). **Boletim do Comando da Aeronáutica**, Brasília, DF, n. 106, 09 jun. 2010.

BRASIL. Ministério da Defesa. Comando da Aeronáutica. Portaria nº 836/DLE, de 2 de maio de 2019. Aprova a Norma que dispõe sobre Correspondência e Atos Oficiais do Comando da Aeronáutica (NSCA 10-2). **Boletim do Comando da Aeronáutica**, Brasília, DF, n. 72, 02 maio 2019.

DEVEREAUX, J. E. **Obsolescence**: A Systems Engineering And management Approach For Complex Systems. 2010. Dissertação (Mestrado em Engenharia Aeroespacial) – Massachusetts Institute Of Technology, Cambridge, Estados Unidos, 2010.

DRAPER, G. Managing Cybersecurity Risks Using a Risk Matrix. **Fort Safe**, Sidnei, Austrália, set. 2019. Disponível em: https://bit.ly/draperrisks. Acesso em: 24 mar. 2020.

HÁRAN, J. M. Microsoft lança patch que corrige vulnerabilidade no SMBv3. **We Live Security**, [s. l.], 16 mar. 2020. Disponível em: http://tiny.cc/smbv3. Acesso em: 23 mar. 2020.

LEAL, A. Governo confirma reajuste de 150% nas diárias de servidores durante a Rio 2016. **Agência Brasil**, Brasília, DF, 15 jul. 2016. Disponível em: https://bit.ly/diariasonix2016. Acesso em: 24 set. 2019.

ORACLE atualiza Java, mas experts dizem que falhas continuam. **G1**, [s. l.], 14 jan. 2013. Disponível em: http://glo.bo/Y5PLf0. Acesso em: 23 mar. 2020.

OLIVEIRA, P. R. M. FAB implanta sistema para unificar os processos de Logística: o SILOMS (Sistema Integrado de Logística de Material e de Serviços). **It4cio**: Case de Sucesso. [s. l.], maio 2016. Disponível em: https://bit.ly/casesiloms. Acesso em: 23 mar. 2020.

OLIVEIRA, W. Gestão da mudança Ágil: rápido e eficiente. **Heflo**, [s. l.], 07 nov. 2016. Disponível em: https://bit.ly/gestaoagilwallace. Acesso em: 23 mar. 2020.

RAJAGOPALA, S.; ERKOYUNCUA, J. A.; ROYA R. Software obsolescence in defence. Global Web Conference. In: CIRP GLOBAL WEB CONFERENCE, 3., 2014. **Anais…**. Cranfield, United Kingdom: Elsevier, 2014. p. 1-5.

RUAS, G. Educação para Redução de Riscos Cibernéticos. Separata de: **Revista Fonte**, Rio de Janeiro, n. 18, p. 10-15, dez. 2017. Disponível em: http://tiny.cc/revfontedez2017. Acesso em: 23 mar. 2020.

WILLIANS, S. Forms 11g: Já começou a se preparar para migrar para o Forms 12c?. **GPO**. [s. l.], 06 abr. 2017. Disponível em: https://bit.ly/silomsnpapi. Acesso em: 23 mar. 2020.