

Edeyson A. Gomes

Metrics and Laws of Software Evolution

Resumo

Um estudo preliminar efetuado com o IBM OS/360 é apresentado para ilustrar a tendência de crescimento do software em função de sua progressão no tempo. A premissa desse estudo centra-se na persistência de problemas que acompanham o software em desenvolvimento ou manutenção. Como resultado, tem-se a hipótese de que o processo de evolução de software seja guiado pelo retorno (*feedback*) positivo e negativo que agem como fatores externos reguladores e estabilizadores do mesmo.

Em função destes fatores, que abarcam a dinâmica organizacional, gerência, sociologia etc., as leis da evolução do software do tipo E, que visam endereçar problemas do mundo real, são descritas e avaliadas.

Embora a semântica de lei retrate uma fórmula geral que enuncia uma relação constante entre fenômenos de uma dada ordem, as leis enunciadas carecem de validações quanto ao domínio de aplicação, sistemas e organizações para tornarem-se instrumento teórico e prático.

A dinâmica dos negócios do mundo real requer que os sistemas do tipo-E sejam continuamente atualizados para manter sua validade. Neste cenário, a busca por melhorias no processo de desenvolvimento e manutenção de software tem sido uma constante na indústria de software. Em consequência disso, visa-se demonstrar que tais processos são centrados em *feedback*. Com este propósito, os autores formulam a hipótese FEAST (FEAST/1) (*Feedback, Evolution and Software Technology*).

Como foco de investigação tem-se a identificação e caracterização dos mecanismos de *feedback* ativos no processo de software, seu impacto nas características deste e métodos para aplicar o conhecimento inferido para melhorá-lo. Duas abordagens são destacadas nesta investigação:

1. Caixa preta, que visa determinar padrões de evolução de software através do estudo de dados quantitativos históricos.
2. Caixa branca, que visa construir modelos dinâmicos de sistemas que reflitam mecanismos de *feedback*, suas propriedades e impactos.

Os resultados do FEAST/1 são baseados nestas duas abordagens aplicadas, como estudo de caso, ao sistema de transações financeiras da Logica Fastwire (FW). Este, um sistema com 8 anos, implantado em uma centena de clientes e com histórico de evolução de 5 anos que abarca 100 versões. Os dados usados para avaliação foram: o número de versão, que estabelece uma cronologia parcial das versões, o tamanho em módulos e o número de módulos.

Como métrica para a avaliação do crescimento dos sistemas sob análise usou-se e a contagem de módulos, em contrapartida ao número de linhas de código ou pontos de função.

O crescimento em módulos do FW foi comparado ao do OS/360, identificando grande similitude. A tendência de crescimento inferida é apontada como consistente com a primeira e sexta leis de evolução de software, embora não as distinga. A saber:

- a. Lei I – Mudança contínua, segundo a qual sistemas do tipo-E devem ser continuamente adaptados para não tornarem-se insatisfatórios. Isso, devido à dinâmica da atividade humana no mundo real que implica na mudança de requisitos e interesses no software.
- b. Lei VI – Crescimento Contínuo, segundo a qual as funcionalidades de sistemas do tipo-E devem ser continuamente incrementadas para manter a satisfação do cliente com o passar do tempo.

A avaliação do crescimento incremental médio do FW, confrontado ao do OS/360, apontou características similares. O padrão cíclico de crescimento observado alicerçou a formulação da terceira e quinta leis:

- a. Lei III – Auto Regulação, segundo a qual o processo de evolução de sistemas do tipo-E é auto regulado com a distribuição do produto.
- b. Lei V – Conservação de Familiaridade, segundo a qual a evolução de sistemas do tipo-E leva à manutenção do controle de seu comportamento por quem interage com ele, sendo que o crescimento excessivo pode diminuir tal controle. O crescimento incremental médio permanece invariante com a evolução do sistema.

A vulnerabilidade da quinta lei é abordada em função da amostra de dados analisada e os fatores reais que causam o padrão cíclico de variação de crescimento, como complexidade e tecnologia. Ainda, aponta-se a incerteza do comportamento comum que deve ser inferido para outros sistemas e outros domínios.

Em sua defesa, explana-se claramente a relação do *feedback* positivo com o crescimento do software e como este, sendo negativo, implica num período de consolidação e estabilização do crescimento. Respectivamente, os pontos altos e baixos delineados no gráfico de crescimento médio dos softwares descritos.

A abordagem caixa branca de investigação do *feedback*, embora ainda incipiente, demonstra dois modelos – IS (*Inverse Square*) e LSL (*Least Squares Linear*) – e aponta o IS com mais aderente aos objetivos do FEAST por refletir o crescimento de complexidade do software em progressão com suas versões.

A progressão de complexidade é alvo da segunda lei de evolução do software, Complexidade Crescente, segundo a qual essa é continuamente incrementada, a menos que algo seja feito para controlá-la ou reduzi-la. O modelo IS subsidia seu enunciado e o da quarta lei, Conservação da Estabilidade Organizacional. Segundo esta, a média da taxa de atividades na evolução de um software do tipo-E é invariante com o tempo. Esta taxa, que se supõe representar o esforço para levar o sistema de uma versão a outra, seria, então, o parâmetro \underline{E} do modelo IS.

Resultados do FEAST/1 apontam para evidências de um padrão de crescimento de software que independem de suas características de desenvolvimento e ambiente, embora requeiram aprofundamento posterior para validar tais conclusões.

Crítica

O texto apresenta uma metodologia detalhada sobre avaliação da evolução de software com base em seu crescimento incremental. Formulada uma hipótese, a FEAST, descrevem-se os modelos de investigação adotados para comprová-la.

Embora a metodologia seja clara, ainda apresenta vulnerabilidades em suas conclusões devido ao limite claro de sua amostragem baseada em dois sistemas: o OS/360 e o FW. Ainda, a métrica considerada na avaliação – número e tamanho de módulos – tem sua defesa fragilizada com a assertiva dos autores que a mesma é usada na ausência de uma melhor e que é uma estimativa inicial.

O subsídio do estudo à enunciação das leis segue um raciocínio lógico coeso, embora se sinta falta de um estudo mais aprofundado. A relação de causa e consequência do *feedback* positivo e negativo para o crescimento do software e sua associação às leis é feita de forma precisa em alguns casos (algumas leis), embora fique notório a necessidade de investigação em outros domínios e tipos de software. Os autores são conscientes disso e apontam a incipiência do estudo em diversos pontos do texto.

Como grande contribuição ressalta-se a abordagem que correlaciona a dinâmica de evolução dos problemas no mundo real com a natureza evolutiva de softwares do tipo-E e como a interação com os usuários implica em seu crescimento. A busca de um padrão na natureza dessa interação pode conduzir a melhorias no processo de desenvolvimento e progressão de software.

Questões para discussão

1. A métrica baseada em quantidade e tamanho de módulos é consistente para o estudo do crescimento incremental do software?
2. A semântica de lei, definindo uma fórmula geral que enuncia uma relação constante entre fenômenos de uma dada ordem, é seguida pelas referidas leis de evolução do software?
3. A amostragem efetuada com o OS/360 e o FW é suficiente para comprovar a constância de fenômenos que causam ou são consequentes da evolução do software?

Referência

Lehman, M.M.; Ramil, J.F.; Wernick, P.D.; Perry, D.E.; Turski, W.M. Metrics and laws of software evolution - the nineties view. Fourth International Software Metrics Symposium, 5-7 Nov. 1997. Proceedings, 1997, p. 20-32.