

PROPOSTA DE METODOLOGIA PARA AVALIAÇÃO DE SOFTWARE COMERCIAL DESTINADO AO PROJETO DE ENGENHARIA DA CONSTRUÇÃO



1. Introdução

A partir da década de noventa do século passado, começaram a surgir no mercado ferramentas de cálculo com o objetivo de agilizar a conceção do projeto de engenharia da construção.

Até ao final da década de setenta os computadores existentes eram enormes, apenas entidades de grande poder económico os podiam adquirir. Na década de oitenta surgiu no mercado o PC, *Personal Computer*, estas pequenas máquinas começaram a ser adquiridas pela generalidade das empresas e em Portugal no final desta década era possível encontrar indivíduos que já possuíam o seu PC.

Na década de noventa, a saída de recém-formados das instituições de ensino superior, fomentou no mercado o aparecimento de empresas de informática dedicadas à conceção de software de acordo com as necessidades do próprio mercado, daí resultando software comercial à medida e software comercial de prateleira (COTS, *Commercial Off-The-Shelf*)^[1].

O software comercial, ao ser utilizado por um grande número de pessoas, atingindo facilmente, no caso do COTS, os milhares, tem condições para evoluir de acordo com as

exigências sistemáticas do próprio mercado, atingindo elevados patamares no cumprimento de requisitos de qualidade, nomeadamente no que concerne à funcionalidade, fiabilidade, usabilidade, manutenibilidade, eficiência, portabilidade e qualidade na utilização.

A utilização de software comercial na área do projeto de engenharia da construção é hoje em dia uma prática absolutamente generalizada. A seleção do software pode tornar-se um processo complexo especialmente naquelas áreas em que existe grande oferta. A utilização de critérios de avaliação bem definidos poderá agilizar o processo e dar maiores garantias no momento da decisão final. Neste documento apresenta-se uma proposta de metodologia para avaliação e comparação de softwares.

2. A evolução da utilização do software, apresentação de um caso

A empresa Top Informática, está no mercado desde 1988, dedica-se à conceção, validação e comercialização de software para o projeto de engenharia da construção. Apresenta-se como líder de mercado desde os anos noventa, tendo iniciado a comercialização do software CYPE desde 1991.

Com base nos seus registos, relativos a vendas, apresentam-se dois gráficos, figura 1 e figura 2, que permitem observar a evolução da utilização do software como ferramenta para o projeto de engenharia da construção, bem como a importância que os softwares específicos de cada especialidade têm adquirido ao longo dos anos.

As instituições de ensino superior estão também cada vez mais empenhadas em disponibilizar para os seus alunos *softwares* comerciais, como meio de preparação para o mercado de trabalho. A análise do gráfico da figura 3 permite observar a evolução da utilização do *software* pelas instituições de ensino superior, no âmbito dos protocolos estabelecidos com a Top Informática.

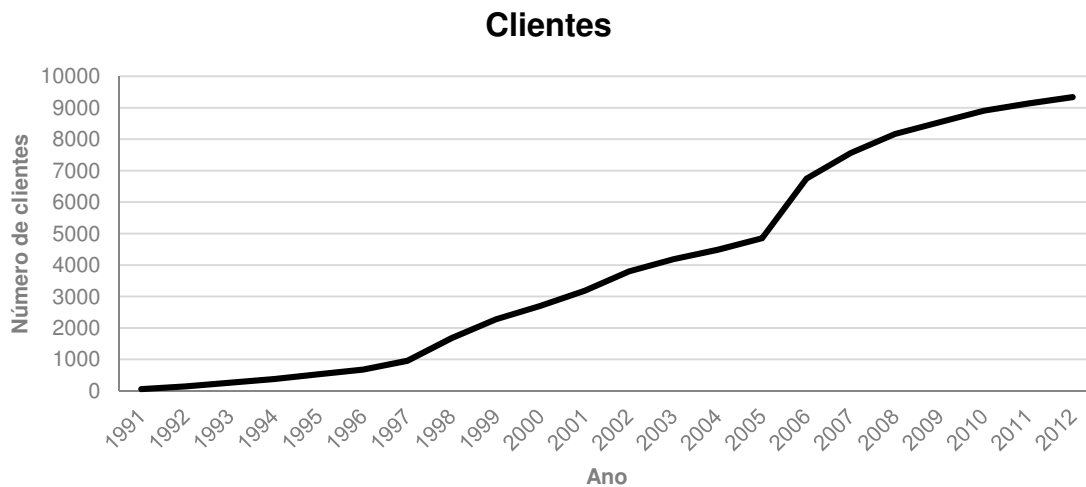


Figura 1 – Evolução do número de clientes CYPE

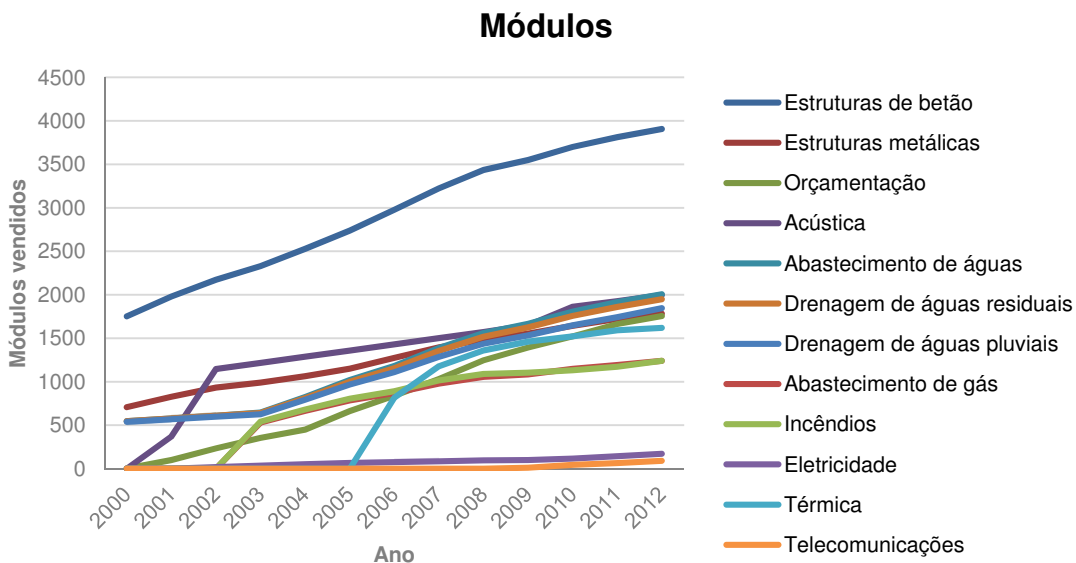


Figura 2 – Evolução do número de módulo vendidos por especialidade

Contactos protocolos

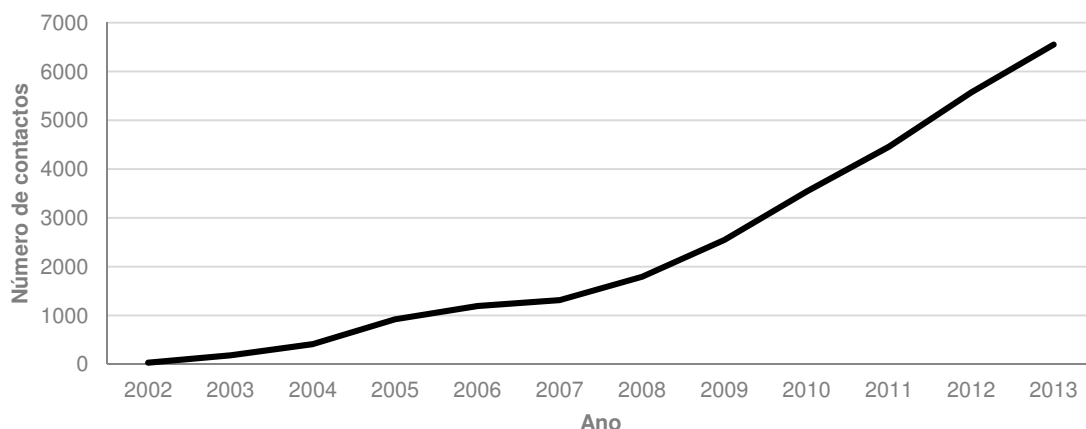


Figura 3 – Evolução do número de contactos de protocolos

Neste gráfico o número de contactos corresponde ao número total de alunos e professores inscritos nos protocolos celebrados entre as várias instituições de ensino superior e a Top Informática. No caso desta empresa o protocolo permite a utilização gratuita do software, bem como o acesso a serviços como assistência técnica, formação, atualizações, entre outros.

3. Certificação de software

O software não poderá em caso algum substituir o engenheiro projetista, independentemente da especialidade de engenharia.

O controlo dos dados e a análise de resultados é da responsabilidade do projetista, no entanto, o software pode ser mais ou menos amigável, ou seja, pode possuir características que facilitam a sua utilização e consequentemente pode permitir uma maior produtividade. Para que tal aconteça o software deve possuir requisitos de qualidade, preferencialmente reconhecidos por entidades competentes.

Pode-se afirmar que a prática da certificação de software não existe a nível mundial. Apesar de existirem normas internacionais disponíveis e entidades internacionais para

certificação, o número de softwares certificado é muitíssimo reduzido. Pode facilmente comprovar-se esta afirmação através da realização de uma pesquisa na Internet.

Em Portugal, de acordo com informação recolhida junto das entidades de certificação de produto, apenas é possível encontrar um software, o *Cypeterm*, certificado de 1 de setembro de 2010 a 30 de novembro de 2013 segundo os requisitos ISO/IEC 25051:2006 + ADENE + CERTIF, Figura 4.



Figura 4 – Marca de certificação ISO/IEC25051:2006 + ADENE + CERTIF

A CERTIF foi, em Portugal, a única entidade que apresentou no seu catálogo o serviço de certificação do produto *software*, com padrão de medição da responsabilidade de entidade reguladora, no entanto, apenas para *software* destinado ao projeto térmico de edifícios. Isto significa que atualmente em Portugal de acordo com os catálogos de serviços disponíveis, não é possível proceder à certificação de nenhum *software* com padrão de medição.

4. Requisitos de qualidade do software

Já foram referidos anteriormente alguns requisitos de qualidade aos quais deve obedecer o software, estes podem ser encontrados em normas de certificação de software como é o caso da ISO/IEC 25051:2006.

Especificam-se seguidamente os requisitos que se consideram de importância relevante para o software de projeto de engenharia da construção, de acordo com a norma referida.

- Funcionalidade

O software deve cumprir a legislação em vigor, relativa à área a que diz respeito, nomeadamente através das suas funções de cálculo descritas em catálogo.

O software deve, dentro da medida do possível, ser constituído por módulos de acordo com as necessidades específicas dos utilizadores.

O software deve ser testado exaustivamente para assegurar aos utilizadores a exatidão dos cálculos efetuados de acordo com as metodologias de cálculo de engenharia aplicáveis.

A arquitetura do software deve permitir a comunicação com outros softwares e, atendendo ao atual estado da arte, deve aplicar tecnologia BIM.

O software deve descrever claramente se possui sistema de proteção, uma vez que a utilização sem o referido sistema constitui crime punível por lei.

- Fiabilidade

O software deve permitir ao utilizador realizar, a qualquer instante, a gravação dos dados. Adicionalmente o software deve efetuar gravações automáticas dos dados. Sempre que se efetue uma nova gravação de dados a anterior deverá ser mantida, sendo possível o seu restauro, em caso de eventual falha.

Ocorrendo uma falha nos recursos do sistema ou rede o software deve permitir recuperar a anterior cópia de segurança da obra e continuar a operar.

- Usabilidade

A introdução de dados deve ser de um modo geral gráfica. Este tipo de interface simplifica o processo da realização de projeto, porque diminui o tempo necessário para a introdução dos dados, bem como o risco de introdução de dados errados.

A partir de dados básicos, o software deve realizar automaticamente cálculos cujos resultados devem servir de inputs para cálculos mais complexos. Este processo diminui a possibilidade de erro humano na introdução de dados.

A interface deve ser intuitiva e conferir ao produto um elevado grau de inteligibilidade, tornando ao mesmo tempo o software prático e de fácil aprendizagem e utilização.

Sempre que necessário o software deve conter ajudas que esclareçam o dado pretendido ou o resultado obtido.

O *software* deve permitir a configuração de menus. Deve também, por exemplo, permitir configurar bibliotecas de materiais e elementos construtivos, entre outras, através de comandos simples de cópia e edição.

O *software* deve conter documentação do utilizador, como manuais, tutoriais, vídeos, etc., com exemplos práticos de aplicação.

O *software* deve ser disponibilizado a utilizadores específicos (investigadores, professores, estudantes e formadores), preferencialmente de forma gratuita. Deve também fornecer versões de avaliação gratuitas para que o software possa ser devidamente analisado pelos potenciais utilizadores.

A usabilidade do *software* deve ser reconhecida publicamente pelos utilizadores, através de estudos de satisfação.

- Eficiência

Devem ser definidos claramente os requisitos mínimos, de hardware e sistema, especialmente os requisitos recomendáveis para o correto funcionamento do software. Estes devem corresponder aos existentes correntemente no mercado.

Os recursos não devem condicionar o desempenho do software. O cálculo deve ser rápido, quando operado num sistema que possua os requisitos mínimos necessários.

- Manutenibilidade

O software deve ter um preço inicial adequado às suas potencialidades.

O software deve ser permanentemente atualizado, através da introdução de sucessivas melhorias ao longo da vida de cada versão.

Deve assegurar também que são contempladas todas as alterações decorrentes de alterações à regulamentação, sistemas operativos, arquiteturas de hardware ou programas interatuantes.

As sucessivas versões do software devem ser submetidas a um rigoroso plano de testes para identificação de eventuais falhas, face às versões anteriores.

O preço das atualizações deve ser tabelado, ou caso tal não seja possível devem ser definidos valores limite.

Deve ser previsto um sistema que reporte à Assistência Técnica eventuais anomalias.

Deve ser assegurada a Assistência Técnica ao software mediante condições claramente definidas.

Deve estar definido o processo de substituição em caso de avaria ou roubo do software e/ou proteção física, caso exista.

- Portabilidade

O software deve poder ser instalado de acordo com os diferentes sistemas operativos previstos. Deve estar definido o processo de instalação do software.

Devem estar definidas as condições de instalação de acordo com as licenças adquiridas para monoposto ou rede.

A reinstalação do software, no caso de atualização, deve permitir manter configurações, como por exemplo barra de ferramentas, periféricos para impressão e bibliotecas.

Deve ser possível copiar as obras com todos os seus dados e também resultados de cálculo de um computador para outro. A cópia deve poder ser feita através da rede ou utilizando um dispositivo de armazenamento. A pasta das obras deve poder estar em qualquer parte do disco ou na rede.

O software deve manter a sua aptidão funcional quando instalado em sistemas com outros softwares.

O software deve cumprir os padrões vigentes no que se refere à portabilidade.

- Qualidade na utilização

O plano de testes relativo a cada versão e atualização do software deve permitir concluir da capacidade do software para a realização do projeto da especialidade em questão, com resultados corretos e completos, por forma a dar resposta à legislação em vigor e práticas construtivas.

Para os modos de falha existentes no software, este deve dispor de avisos que alertem o utilizador para os erros de operação.

As consequências negativas das falhas do software não deverão ser relevantes.

A generalidade dos utilizadores deve declarar-se satisfeita ou muito satisfeita com o desempenho global do software.

5. Metodologia de avaliação de *software*

A presente metodologia é uma proposta para avaliação de *software* de acordo com os requisitos de qualidade especificados no ponto 4. É possível definir requisitos adicionais, para os quais se poderá aplicar esta mesma metodologia.

Considere-se o requisito “Funcionalidade” definido no ponto 4. Designe-se por F_c .

Considerem-se para este requisito as seguintes características:

F_{c1} O *software* cumpre a legislação em vigor, para a área a que diz respeito

F_{c2} O *software* é constituído por módulos de acordo com as necessidades do utilizador

F_{c3} Existem evidências dos testes realizados que comprovem a exatidão dos cálculos

F_{c4} O *software* permite a comunicação com outros softwares

F_{c5} O *software* aplica a tecnologia BIM

Para este requisito definiram-se 5 características. Igual procedimento pode ser aplicado a cada um dos restantes requisitos, podendo o número de características de cada requisito tomar o valor de “n”.

Deste modo podem-se estabelecer as seguintes expressões:

$$F_c = \sum_{i=1}^n \alpha_i F_{ci} \quad (1)$$

Em que:

α_i Coeficiente de ponderação da característica F_{ci}

$$0 < \alpha_i \leq 1, \quad \sum_{i=1}^n \alpha_i = 1$$

F_{ci} Pontuação atribuída à característica “i” do requisito “Funcionalidade”

$$F_b = \sum_{i=1}^n \beta_i F_{bi} \quad (2)$$

Em que:

β_i Coeficiente de ponderação da característica F_{bi}

$$0 < \beta_i \leq 1, \quad \sum_{i=1}^n \beta_i = 1$$

F_{bi} Pontuação atribuída à característica “i” do requisito “Fiabilidade”

$$U_s = \sum_{i=1}^n \gamma_i U_{si} \quad (3)$$

Em que:

γ_i Coeficiente de ponderação da característica U_{si}

$$0 < \gamma_i \leq 1, \quad \sum_{i=1}^n \gamma_i = 1$$

U_{si} Pontuação atribuída à característica “i” do requisito “Usabilidade”

$$E_f = \sum_{i=1}^n \delta_i E_{fi} \quad (4)$$

Em que:

δ_i Coeficiente de ponderação da característica E_{fi}

$$0 < \delta_i \leq 1, \quad \sum_{i=1}^n \delta_i = 1$$

E_{fi} Pontuação atribuída à característica “i” do requisito “Eficiência”

$$M_n = \sum_{i=1}^n \rho_i M_{ni} \quad (5)$$

Em que:

ρ_i , coeficiente de ponderação da característica M_{ni}

$$0 < \rho_i \leq 1, \quad \sum_{i=1}^n \rho_i = 1$$

M_{ni} , pontuação atribuída à característica “i” do requisito “Manutenibilidade”

$$P_r = \sum_{i=1}^n \mu_i P_{r,i} \quad (6)$$

Em que:

μ_i Coeficiente de ponderação da característica $P_{r,i}$

$$0 < \mu_i \leq 1, \quad \sum_{i=1}^n \mu_i = 1$$

$P_{r,i}$ Pontuação atribuída à característica “i” do requisito “Portabilidade”

$$Q_u = \sum_{i=1}^n v_i Q_{u,i} \quad (7)$$

Em que:

v_i Coeficiente de ponderação da característica $Q_{u,i}$

$$0 < v_i \leq 1, \quad \sum_{i=1}^n v_i = 1$$

$Q_{u,i}$ Pontuação atribuída à característica “i” do requisito “Qualidade na utilização”

A pontuação a atribuir a cada característica poderá obedecer ao seguinte critério:

mau = 1; razoável = 2; bom = 3; excelente = 4.

A pontuação, p , obtida pelo *software* será dada pela expressão:

$$p = (F_c + F_b + U_s + E_f + M_n + P_r + Q_u) / 7 \quad (8)$$

Na expressão (8), o somatório das pontuações ponderadas obtidas para cada requisito é dividido por 7, sendo este o número de requisitos especificados no ponto 4, no entanto, outros requisitos poderão ser definidos, aplicando esta mesma metodologia. Também pode ser criada a hipótese de atribuir pesos a cada requisito, uma vez que estes podem ter entre si distinta importância.

6. Conclusão

A utilização de *software* comercial para o projeto de engenharia da construção tem sofrido uma enorme evolução desde a década de noventa até à atualidade, conseqüentemente tem também surgido no mercado uma enorme oferta para cada especialidade do projeto. A seleção de *software* pode tornar-se num processo complexo, especialmente por ser difícil ao utilizador identificar claramente quais os requisitos de qualidade que este deve cumprir. A certificação do *software* como meio de reconhecimento do cumprimento de requisitos facilitaria o processo de seleção, no entanto, o *software* certificado é praticamente inexistente no mercado. Este documento orienta o utilizador para a identificação de requisitos e apresenta uma proposta de metodologia de avaliação do *software*. A aplicação desta metodologia a vários *softwares* permite obter a respetiva pontuação, servindo esta de termo de comparação entre eles, podendo, assim, o utilizador selecionar o *software* mais eficiente para a sua atividade.

7. Referências bibliográficas

- (1) ISO/IEC 25051:2006. Software engineering — Software product Quality Requirements and Evaluation (SQuaRE) — Requirements for quality of Commercial Off-The-Shelf (COTS) software product and instructions for testing.
- (2) ISO/IEC 9126-1:2001. Software engineering — Product quality — Part 1: Quality model.
- (3) ISO/IEC TR 9126-2:2003. Software engineering — Product quality — Part 2: External metrics.
- (4) ISO/IEC TR 9126-4:2004. Software engineering — Product quality — Part 4: Quality in use metrics.
- (5) ISO 9127 1988. Information processing systems — User documentation and cover information for consumer software packages.
- (6) ISO/IEC 90003:2004. Software engineering — Guidelines for the application of ISO 9001:2000 to computer software.