

Reutilização de Software

João Pedro Dias¹

Universidade de Évora, Évora, Portugal
m42055@alunos.uevora.pt

Resumo Atualmente, destaca-se um aumento na utilização quotidiana de sistemas de software. O referido incremento na procura espoleta uma necessidade de reinvenção na área de Engenharia de Software, que se manifesta na exploração de novas técnicas de desenvolvimento e melhoria de metodologias pré-existentes. No presente artigo, procede-se a uma abordagem referente ao conceito de reutilização de software, incluindo a análise de vantagens e desvantagens intrínsecas a esta técnica. Nesta linha de pensamento, os sistemas baseados em serviços, cada vez mais procurados no âmbito de soluções tecnológicas, destacam-se como um possível exemplo de reutilização de software. Como tal, pretende-se descrever o seu funcionamento bem como, estabelecer uma relação com a técnica previamente mencionada. Em última análise, com intuito de garantir a funcionalidade de sistemas com esta tipologia, é crucial estabelecer uma relação entre os dois conceitos enunciados e o processo inerente ao teste de software.

Keywords: Engenharia de Software · Reutilização · Serviços · Testes de Software

1 Introdução

Apesar da intensificação no progresso tecnológico registado nas últimas décadas, o conceito de software permanece bastante abstrato. Em geral, software corresponde a um conjunto de instruções indicativas da tarefa a executar pelo hardware do computador [1], englobando qualquer programa, procedimento ou rotina associada a operações ao nível de um sistema de computação. Nesta perspetiva, surge o conceito de Engenharia de Software, uma disciplina, na área da engenharia, que envolve todos os aspetos relacionados com produção de software. Os temas em estudo no presente artigo, nomeadamente, reutilização de software, software baseado em serviços e ainda testes de software, enquadram-se nesta área.

Em primeiro lugar, o foco será a reutilização de software. Esta temática, tem vindo a adquirir maior relevância na sociedade contemporânea uma vez que, se apresenta como um dos métodos de desenvolvimento mais rentáveis neste ramo. A principal característica das técnicas de reutilização de software prende-se com a velocidade, no entanto, no decorrer deste documento outras menos significativas serão, adicionalmente, referidas. Na tentativa de descodificar se

a reutilização de software se apresenta como uma técnica futuramente viável, no contexto de produção de software, será realizada uma averiguação das suas vantagens e desvantagens.

Enquadrado na reutilização de software, verifica-se a existência de outro conceito, também desenvolvido neste artigo, designado por software baseado em serviços. Segundo [2], o desenvolvimento deste tipo de software tem-se equiparado ao desenvolvimento de software orientado a objetos. De acordo com esta metodologia, o sistema que utiliza o serviço encarrega-se, principalmente, da lógica de integração com o mesmo, o que se revela uma das vantagens inerentes ao desenvolvimento de software baseado em serviços, pois qualquer processamento mais complexo é, eventualmente, delegado ao serviço em si. Esta característica, aliada à possibilidade de reutilização de serviços, possibilita a criação de sistemas leves mas, com capacidade de executar tarefas tão ou mais complexas que um sistema tradicional.

No decorrer do presente artigo, além de se proceder à análise dos pontos fortes associados aos dois conceitos supramencionados, será ainda introduzido o tema relativo a testes de software, responsáveis por garantir a qualidade e consistência do sistema. De acordo com [3], estima-se que, atualmente, durante um projeto de construção de software típico, a divisão de tempo útil consista em: 50% para desenvolvimento e 50% para testar o sistema. A dimensão da parcela temporal destinada para testes de software permite destacar a importância deste procedimento. Por último, pretende-se alcançar a percepção do conceito geral de testes de software bem como, compreender a sua serventia para software baseado em serviços e, de que modo, os referidos testes contribuem para a consistência e durabilidade do sistema.

2 Reutilização de Software

A reutilização de software corresponde a uma estratégia de desenvolvimento de software progressivamente mais popular. O incremento da procura por software simples e de baixo custo constitui a principal razão para a recente adoção desta estratégia por parte de programadores de sistemas. Adicionalmente, a ambição por software com melhor qualidade e tempo de entrega mais curto traduz-se numa utilização mais frequente desta técnica [2].

O interesse em reutilizar software também está associado ao aumento na disponibilidade de acesso a software livre, mais conhecido por software open-source. Como tal, é possível aferir que a participação recorrente de um programador em projetos open-source se associa a uma maior probabilidade de utilização de software reutilizável [4]. Este tipo de software coaduna-se com o conceito relativo ao desenvolvimento em comunidade, evidenciando como principal objetivo a implementação de software com maior qualidade e, que pode ser reutilizado por qualquer pessoa.

Existem diversas técnicas que permitem reutilizar software, podendo ser separadas em quatro categorias distintas: reutilização de sistemas, reutilização de aplicações, reutilização de componentes e reutilização de objetos ou funções.

No entanto, antes de seleccionar a mais adequada, verifica-se a necessidade de averiguar o contexto do projeto e rever alguns fatores chave abordados em [2]. Estes fatores são responsáveis por transmitir ao programador se é conveniente reutilizar software e, em caso afirmativo, qual a técnica mais indicada no âmbito do projeto a desenvolver. O prazo para desenvolvimento constitui um dos fatores mais significativos dado que, caso o período temporal para entrega da solução seja escasso, proceder à implementação de um sistema de raiz poderá gerar atrasos indesejáveis como tal, é preferível recorrer à reutilização. Além disso, a capacidade da equipa de desenvolvimento e o domínio da aplicação são fatores a ter em consideração, com intuito de evitar sobrecarga da equipa ou plataforma de desenvolvimento.

Quando a reutilização de software se trata da solução mais adequada, o passo seguinte tem por base a escolha da melhor técnica para o efeito. O leque é bastante variado e abrangente [2], estendendo-se desde a técnica de sistemas de sistemas, que consiste em integrar dois ou mais sistemas formando um novo sistema, até à reutilização de padrões de desenho ou de arquitetura, onde o contexto de reutilização se revela mais abstrato. Considerando todas as técnicas disponíveis, a utilização de frameworks de aplicação, devido à sua capacidade de fornecer ao programador uma lista de funcionalidades genéricas que podem ser aplicadas com intuito de facilitar o desenvolvimento, corresponde à mais utilizada.

A utilização desta estratégia em Engenharia de Software proporciona benefícios significativos no desenvolvimento de novos sistemas e aplicações, apurando-se como principal vantagem a velocidade com que se implementa a própria solução. Ao reutilizar software é viável reduzir o tempo necessário para desenvolvimento visto que, o programador já possui, a priori, a base para a solução, sendo apenas necessário efetuar uma adaptação da mesma. Deste modo, verifica-se a possibilidade de realizar uma gestão mais eficiente na alocação da equipa de desenvolvimento, com necessidade de menor número de programadores decorrente da redução no grau de complexidade de maior parte das tarefas [2].

Outro dos pontos fortes que a abordagem mencionada acarreta, prende-se com a redução de custos de desenvolvimento visto que, é expetável uma redução de software implementado de raiz. A referida redução agrada a parte substancial das empresas de software pois, para além de reduzir o risco de implementação, proporciona um aumento na margem de lucro do projeto. Este fator é bastante relevante uma vez que, na maioria dos projetos os custos de desenvolvimento são muito elevados. Por outro lado, os custos de manutenção tendem a sofrer um aumento dado que, o risco de todo o sistema colapsar devido a falha de um só componente é multiplicado pelo número de vezes em que o mesmo é reutilizado no sistema.

O facto do resultado final se traduzir num sistema mais homogêneo também se apresenta como uma particularidade interessante para quem procura este tipo de desenvolvimento de software. Ao reutilizar software ocorre um aumento de semelhanças no sistema, facilitando o cumprimento de normas standard e o

processo de transferência de conhecimento, caso um novo elemento se junte à equipa de desenvolvimento.

Apesar desta tipologia de desenvolvimento proporcionar um número acentuado de benefícios relacionados com velocidade, custos e homogeneidade, existem alguns desafios associados à manutenção, suporte e adaptabilidade [2]. O desafio relacionado com manutenção não se reflete, exclusivamente, a nível financeiro estando, também, presente em contexto técnico. O facto da totalidade do sistema estar dependente de um ou mais componentes reutilizados, implica a necessidade de atenção redobrada a qualquer tipo de falha que os possa afetar bem como, a novas atualizações que possam não ser suportadas pelo sistema, sendo impreterível uma nova adaptação.

O desafio associado ao conceito de suporte correlaciona-se com situações em que o software reutilizado não é da autoria do programador como tal, poderá verificar-se uma dependência de terceiros na eventualidade de surgirem problemas mais complexos. Esta possibilidade deve ser considerada no momento da escolha do software a reutilizar, principalmente, quando o objetivo é garantir a durabilidade de um sistema.

Por último, destaca-se ainda a problemática inerente à adaptação de componentes reutilizáveis ao próprio sistema. Como a maioria do software reutilizável é desenhado para situações genéricas, verifica-se uma situação adversa quando o mesmo é aplicado em projetos com requisitos específicos.

De acordo com [5], podemos concluir que a abordagem de reutilização de software acarreta bastantes benefícios para organizações que desenvolvem software, pois melhora a produtividade e a qualidade de software. Por outro lado, esta tarefa também é considerada um tanto complexa, principalmente, quando se lida com legacy software.

3 Software baseado em serviços

Com intuito de correlacionar este tema com o anterior é fundamental aferir o conceito de software baseado em serviços. Este tipo de software consiste numa técnica delineada para criar sistemas distribuídos, na qual os componentes correspondem a serviços isolados suscetíveis a acesso por parte de outros programas, proporcionando operações de consulta e atualização de recursos externos [2]. Dado que estes serviços podem ser reutilizados, é possível a interação de diversos sistemas com o mesmo serviço sem qualquer objeção. Um exemplo prático desta abordagem assenta em serviços web. Este tipo de sistemas permite que qualquer organização disponibilize os seus recursos a outros programas através da web, requerendo apenas a criação de uma interface que defina todas as operações passíveis de ser executadas.

Perante esta definição é plausível associar o desenvolvimento de software baseado em serviços à reutilização de software uma vez que, um serviço em si constitui um componente isolado que pode ser reutilizável. Além disso, o facto de um serviço ser independente da linguagem e da plataforma facilita a sua

reutilização. Esta característica é essencial para organizações que desenvolvam software baseado em serviços pois potencia a cooperação entre estas entidades.

Para além de possibilitar a reutilização de software, existem mais vantagens que incentivam a implementação de software baseado em serviços [2]. A principal vantagem a referir prende-se com o facto de os serviços poderem ser disponibilizados por uma entidade externa ou interna à organização que desenvolve o sistema, aumentando assim o potencial do mesmo em termos de funcionalidades oferecidas. Outro ponto forte desta metodologia de desenvolvimento consiste em permitir que uma aplicação, que usufrua de um ou mais serviços, seja, totalmente desenvolvida antes de ser estabelecida a integração com o serviço. Complementarmente, qualquer alteração posterior do serviço não implica redesenhar a totalidade da aplicação, existindo, também, a possibilidade de desenvolver uma aplicação mais pequena e que não consuma tantos recursos físicos.

A nível financeiro este tipo de desenvolvimento revela vantagens tanto para quem desenvolve os serviços como para quem os utiliza. Explorar novas oportunidades de negócio é bastante atrativo e pode resultar em lucro para a entidade que desenvolve o serviço. Na ótica do utilizador, a vantagem consiste em pagar pelos serviços apenas quando usufrui dos mesmos, verificando-se a inexistência de um valor de pagamento fixo.

Os serviços fornecidos dividem-se em três categorias fundamentais [2]: serviços utilitários, serviços funcionais e serviços de coordenação. Os serviços utilitários prestam funções utilitárias e genéricas, entre as quais, a conversão monetária. Os serviços que referenciam funcionalidades mais específicas do negócio designam-se como funcionais. Por último, os serviços de coordenação especificam funcionalidades que envolvem mais do que uma entidade, por exemplo, um módulo de compras online.

Uma vez definido o conceito de software baseado em serviços bem como, o próprio conceito de serviço, é possível explorar a arquitetura deste tipo de sistemas. Ao implementar um serviço e especificar a sua interface, a organização responsável pelo mesmo publica a informação num repositório. Após a sua publicação, o serviço fica disponível para quem o quiser requisitar. Resta apenas estabelecer a ligação entre a aplicação concebida e o serviço mais adequado [2]. De modo a evitar incompatibilidades entre diferentes serviços e aplicações existem standards fundamentais, previamente definidos, nomeadamente o SOAP, standard que procede à troca de mensagens entre serviços, e o WSDL (Web Service Description Language), standard que define as interfaces dos serviços e como devem ser realizadas as respetivas ligações.

Atualmente esta standardização é considerada por muitos substancialmente exigente, pelo que diversas organizações recorrem a serviços web RESTfull por serem considerados mais leves. Este tipo de serviços permite ao utilizador solicitar acessos e manipular representações textuais de recursos da web. Para tal, utiliza-se um conjunto predefinido de operações sem estados [6].

4 Testes de software

Na área de engenharia de software é viável separar o ciclo de vida de um sistema de software em várias etapas, mais concretamente, sete [fig.1]. O ciclo inicia-se com o planeamento da solução a desenvolver, seguindo-se a análise de requisitos por parte das entidades envolvidas no projeto. Finalizada a estruturação do projeto o próximo passo consiste em desenhar a solução que se pretende implementar. Após a solução estar desenhada, os programadores podem iniciar a fase de implementação, que corresponde à passagem do desenho para código. Por último, resta testar a solução implementada e caso tudo esteja de acordo com o requerido, disponibiliza-se a solução aos utilizadores e procede-se para a fase de manutenção. Este ciclo repete-se sempre que seja necessário incrementar a solução ou corrigir algum problema posteriormente detetado.

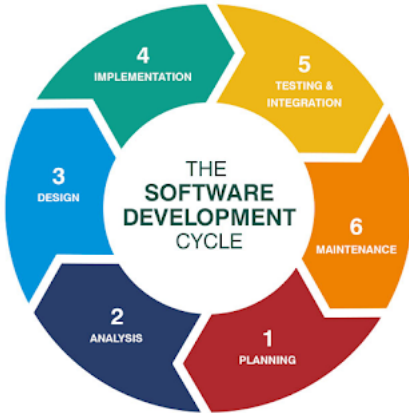


Figura 1. Representação ilustrativa do ciclo de desenvolvimento de um sistema de software

Neste capítulo pretende-se que o foco incida apenas numa das fases descritas, a de testes. O processo de testar software é definido em [3], como a execução de um programa com intuito de detetar erros. A submissão de um programa a testes visa melhorar a sua qualidade e potenciar a confiança no mesmo. Adicionalmente, é importante frisar que, geralmente, os testes de software são realizados com dados artificiais, revelando a presença de erros mas, nunca a sua ausência. Posto isto, os testes de software apresentam como finalidade demonstrar que o software vai de encontro aos requisitos (testes de validação) e que o seu comportamento é igual ao especificado na fase de planeamento (testes de defeitos) [2].

Existem quatro etapas essenciais durante o processo de testes. A primeira corresponde à especificação dos casos de teste, onde se enuncia a funcionalidade alvo de teste e o respetivo resultado esperado. O passo seguinte salienta como objetivo a preparação dos dados que serão usados nos casos de teste. Estando

na posse dos casos de teste e respetivos dados, procede-se à execução dos testes. As sucessivas execuções originam resultados que devem ser, posteriormente, comparados com os resultados esperados de modo a aferir conclusões sobre o sistema [2].

Os testes de software podem dividir-se em distintas categorias, desde testes de stress a testes de sanidade ou compatibilidade [3]. No entanto, durante o desenvolvimento de software comercial, é quase imperativo executar três tipos diferentes de testes [2], nomeadamente, testes de desenvolvimento, testes de release e testes de utilizadores. Os testes de desenvolvimento baseiam-se em testar uma correção ou novo incremento no sistema imediatamente após à implementação do mesmo. Normalmente, nestes testes participa toda a equipa de desenvolvimento. Na segunda categoria de testes referida pretende-se testar todas as funcionalidades que irão ser lançadas na nova versão do sistema, momentos antes desta ser disponibilizada ao utilizador final. Em relação aos testes de utilizadores os mesmos são executados pelo utilizador final do sistema na sua própria plataforma e simulando as condições finais do produto.

4.1 Testes de software baseado em serviços

O software baseado em serviços é um tipo específico de software, no qual se revela essencial a realização de uma carga mais extensa de testes do que num software tradicional. Tal deve-se ao facto de se tratar de um tipo de software onde o elemento confiança é extremamente valorizado, sendo que uma forma eficaz de potenciar a confiança por parte do cliente consiste em demonstrar que o produto foi alvo de um processo de teste minucioso [7].

Um software baseado em serviços implica a necessidade de especificar os vários papéis envolvidos visto que, nem todos possibilitam o teste do sistema nos mesmos moldes. O programador do serviço é o único que, ao ter acesso ao código, o pode testar com base nesse conhecimento, executando testes mais estruturais. Por outro lado, o responsável pela integração do serviço, não consegue realizar testes estruturais, pois não possui acesso ao código do serviço, ficando assim restrito a testes de âmbito funcional [7]. As abordagens de teste para este tipo de software diferem consoante a perspectiva do programador.

Além disso, é relevante denotar que existem várias limitações a superar durante o processo de testes sobre software baseado em serviços. O principal desafio corresponde à limitação de observação do código fonte uma vez que, apenas o fornecedor do serviço tem controlo sobre o mesmo. O custo resultante da realização de testes também se pode manifestar como um grande desafio dado que, geralmente, a utilização de um serviço acarreta sempre custos acrescidos. Por último, o dinamismo que a solução oferece caso seja utilizado mais do que um serviço pode causar problemas senão estiver bem definido que serviços vão ser recrutados durante a execução de um qualquer processo [7].

5 Conclusão

Analisando o conteúdo do presente artigo, confirma-se a ideia inicial de que os dois conceitos de maior destaque, reutilização de software e software baseado em serviços, se apresentam como duas mais valias na área de Engenharia de Software, registando um aumento progressivo de popularidade a nível profissional, académico ou recreativo.

É possível concluir que a maior vantagem associada à reutilização de software corresponde à celeridade no desenvolvimento de software, libertando tempo útil para outras tarefas e, consequentemente, permitindo a redução de custos na implementação da solução. Por outro lado, aferem-se algumas desvantagens relativamente a esta abordagem de desenvolvimento, mais concretamente, o incremento da dependência entre componentes que resulta no aumento do risco de um sistema colapsar, caso um desses componentes falhe.

Com base no tópico incidente sobre a temática de software baseado em serviços destaca-se o facto deste conceito estar intrinsecamente relacionado com reutilização de software visto que, um só serviço pode ser reutilizado por um ou mais serviços na mesma instância.

Em última análise, a exploração do processo de testes de software revelou o seu papel crucial no ciclo de desenvolvimento de um sistema. Quando aplicados a software baseado em serviços, os referidos testes provam ser essenciais para manter a consistência do sistema e transmitir confiança aos utilizadores dos serviços.

Referências

1. Britannica <https://www.britannica.com/technology/software>. Last accessed 14 Apr 2020
2. Sommerville I.: Software engineering, tenth edition. Pearson Education, Harlow (2019).
3. Myers, G., Sandler, C., Badgett, T.: The art of software testing. 3rd edn. John Wiley & Sons, Hoboken, N.J (2012).
4. Sojer, M., Henkel, J.: Code Reuse in Open Source Software Development: Quantitative Evidence, Drivers, and Impediments. *Journal of the Association for Information Systems*. 11, 868-901 (2010).
5. Younoussi S., Roudies O.: All about software reusability: a systematic literature review. *Journal of Theoretical and Applied Information Technology*, 1-12 (2014).
6. REST API Tutorial, <https://restfulapi.net>. Last accessed 13 Apr 2020
7. Bozkurt, M., Harman, M., Hassoun, Y.: Testing and verification in service-oriented architecture: a survey. *Software Testing, Verification and Reliability*. 23, 261-313 (2012).