Se você é de TI, tenho certeza que já ouviu diversas vezes a expressão “sistema legado”. Porém, a não ser que você tenha realizado uma pesquisa por conta própria, duvido que alguém tenha se dado ao trabalho de lhe explicar o que isso significa. Como ocorre com diversos jargões, muitas pessoas os usam sem saber seu exato significado e quem ouve fica constrangido de dizer que não sabe exatamente o que aquilo significa. Assim surgem as lendas. Afinal, todos nascemos sabendo algo tão óbvio, não é mesmo?

**Definindo um sistema legado**

Mas o que é um *sistema legado*? Alguns dizem que é um sistema antigo, outros que é um sistema recebido de terceiros e há quem diga que é um sistema que não pode ser alterado. Na verdade, não existe uma definição única do que é um sistema legado, mas existe um conjunto de indicadores que devemos analisar. Vejamos alguns:

**Tempo de Vida**

Um sistema “velho”, desenvolvido há um tempo relativamente grande, não é necessariamente um legado. O tempo que um sistema tem pode ser um indicador inicial de alerta, mas não define um sistema como legado.

**Utilidade**

Um sistema velho que não é mais usado não é considerado um legado. Um bom indicador de um sistema legado é quando o mesmo ainda é essencial para o negócio, isto é, não se consegue abandoná-lo.

**Tecnologias e Hardware Obsoletos**

Um terceiro indicador de um sistema legado é a sua obsolescência. Alguns sistemas funcionam em hardware antigo, principalmente *mainframes*, cujos componentes são extremamente caros. Linguagem de programação, ferramentas de desenvolvimento, protocolos, bancos de dados e formatos de arquivos depreciados também são itens importantes.

**Dificuldade de Manutenção**

Um outro indicador é se os desenvolvedores originais do sistema já se aposentaram ou trocaram de emprego e é difícil encontrar mão-de-obra para manutenção. Na verdade, se “colocar a mão num vespeiro” é uma metáfora válida para mexer em um sistema, quase certamente você possui um legado em mãos.

**Documentação**

Alguns autores afirmam que uma característica de sistemas legados é a falta de documentação. Discordo completamente. Se fosse assim praticamente todos os sistemas modernos também seriam legados.

Entretanto, se ninguém conhece as regras de negócio implementadas em um sistema e não é possível extraí-las da implementação de alguma forma, isso pode ser um indicador de um legado.

Evouçao do sistema legado

Evolução de software A vida útil de um sistema não acaba com a realização de sua entrega (PFLEEGER, 2004). Um software, durante o seu ciclo de vida, necessita de evolução contínua. Podem surgir erros e falhas, mudanças nas regras de negócios da empresa, novos requisitos, etc.. A evolução de software é importante porque as organizações são muito dependentes dos sistemas que investiram e que utilizam (SOMMERVILLE, 2011). A evolução de software está diretamente relacionada ao conceito de software legado. Um software pode ser considerado como legado quando encontra-se ainda em operação, foi desenvolvido no passado e sofreu modificações ao longo do tempo sem que recebesse melhorias estruturais e de forma sistemática (De Lucia; FASOLINO; POMPELLA, 2001). A preocupação com o ciclo de vida de um sistema é foco de estudos desde a década de 60. Aplicações legadas foram desenvolvidas a décadas atrás e sofreram várias modificações a fim de se adequarem a mudanças de requisitos e de regras de negócio. Geralmente quando se fala em software legado a primeira característica que se destaca é a má qualidade. Aplicações legadas geralmente possuem código complexo, documentação pobre ou inexistente, ausência de registros de testes e o controle de modificações não é registrado (PRESSMAN, 2011). Programas legados muitas vezes são indispensáveis as organizações que os utilizam, pois as mesmas são dependentes das informações ali armazenadas e manipuladas. Quando tal sistema necessita de mudanças contínuas e significativas, deve-se analisar a 30 Capítulo 2. Fundamentação Teórica viabilidade em seguir realizando manutenções ou então descartar o sistema e construir um sistema novo para substituí-lo. Pfleeger (2004) sugere algumas perguntas como instrumento de avaliação neste caso. O custo de manutenção é muito alto? A confiabilidade do sistema é inaceitável? O sistema não pode mais se adaptar a mudanças adicionais dentro de um período de tempo razoável? O desempenho do sistema ainda está fora das restrições prescritas? As funções do sistema têm utilidade limitada? Outros sistemas fazem o mesmo trabalho melhor, mais rápido ou gerando menos custos? O custo de manutenção do hardware é muito alto, a ponto de justificar sua substituição por um hardware novo e mais barato?(PFLEEGER, 2004, p. 384). Manny Lehman, em seus estudos, observou o comportamento do crescimento e evolução de vários sistemas. A partir desta observação, resumiu seus estudos em 05 leis, conhecidas como Leis da evolução de software (PFLEEGER, 2004). A primeira delas se relaciona à mudança contínua. O software em uso, ou irá passar por mudanças contínuas, ou irá, de forma progressiva se tornando menos útil. Sommerville (2011, p. 169) acrescenta: “A primeira lei afirma que a manutenção de sistema é um processo inevitável”. A segunda lei versa sobre o aumento da complexidade. Ao passo que um sistema passa por manutenções, sua estrutura vai aos poucos sendo degradada e sua complexidade aumenta. Sommerville (2011) considera necessárias as manutenções preventivas, a fim de aprimorar continuamente a estrutura do software, além de simplesmente adicionar funcionalidades. Consequentemente, haverão custos adicionais para a realização de tal manutenção. A terceira lei indica que a dinâmica da estrutura definida inicialmente para o sistema influencia nas mudanças, podendo em alguns casos até restringir modificações em um sistema. Conforme o sistema vai crescendo, vai ficando mais complexo e difícil de ser compreendido, dificultando as modificações. Sommerville (2011) considera que pequenas mudanças graduais são mais viáveis, pois evitam a redução de confiabilidade do sistema. Com a realização de grandes mudanças, há uma tendência de surgirem muitos defeitos. A quarta lei de Lehman é relativa a estabilidade organizacional. Lehman considera que, em grandes projetos, a ocorrência de mudanças de recursos ou de pessoal praticamente não afeta a evolução do sistema a longo prazo. Na quinta lei, Lehman relaciona os incrementos do sistema juntamente com o surgimento de defeitos. Ao adicionar uma nova funcionalidade no sistema, também são adicionados proporcionalmente e de forma inevitável novos defeitos. Sommerville (2011) afirma que desta forma é necessário a cada release de sistema um novo release apenas para 2.2. Evolução de software 31 correção de defeitos. Além disso, ao se realizar grandes incrementos de funcionalidade há a necessidade de se haver um orçamento prevendo correção de defeitos.

Manutenção

Em relação a forma em que se realiza a manutenção de um sistema, Sommerville (2011) define três tipos diferentes de manutenção de software. O primeiro é o de manuten- ção para reparo de defeitos de Software, cujo objetivo é a correção de erros, que podem ser de codificação (de custo mais barato), de projeto (mais caros por envolver reescrita de componentes) ou de requisitos (onerosos por envolverem o reprojeto do sistema existente). O segundo tipo de manutenção está relacionado a questões de adaptação de um software a um ambiente, que é necessária quando se altera, por exemplo, o hardware, o sistema operacional ou quando outro software de apoio muda. O terceiro tipo de manutenção envolve a modificação do sistema com o objetivo de adição ou modificação de funcionalidades, que ocorre pela alteração de requisitos ou em função de mudanças de negócio da organização. Em relação ao terceiro tipo de manutenção de um sistema, Pfleeger (2004) exemplifica que ela pode ser necessária por dois tipos de necessidades. A primeira é através de uma necessidade do cliente, motivada por mudanças nas regras de negócio da empresa. A segunda é relativa a natureza do sistema, que é sujeito a fatores externos. Um exemplo deste caso pode ser um sistema que calcula deduções em folha de pagamento para uma empresa. Estes cálculos dependem de leis municipais, estaduais e federais e havendo modificações em alguma destas leis ou caso a empresa se mude para outro município ou abra uma filial, o sistema pode necessitar de manutenção, mesmo funcionando perfeitamente. Em relação a evolução de sistemas legados, Sommerville (2011) descreve quatro estratégias pelas quais a organização que utiliza o sistema pode optar. A primeira delas é a de descartar o sistema, que deve ser escolhida quando o sistema não contribui de forma efetiva com os processos de negócio da empresa. A segunda opção é a de não alterar o sistema e prosseguir realizando manutenções regulares, que é a melhor escolha quando o sistema é necessário para a empresa, é estável e há poucas solicitações de mudanças. Como terceira alternativa, a empresa pode optar por realizar a reengenharia do sistema e aprimorar sua facilidade de manutenção, que é recomendada quando o sistema já está degradado devido as regulares mudanças e quando ainda há necessidades de mudanças. A última opção é a de substituir o sistema atual por um novo sistema, que é a melhor escolha quando há limitações de hardware no sistema antigo que impossibilitem que o mesmo siga em operação ou quando o custo para a reengenharia do sistema legado é muito superior ao custo de substituição do sistema

Reengenharia

Especificações do sistema->projeto de implementação->novo sistema

Sistema de software existente->compreensão e transformação-> sistema de reengenharia