

INTEGRAÇÃO DE SISTEMAS. LEGADOS

CONVERSA INICIAL

Nesta aula, no primeiro tema, explicaremos como um sistema se degrada ao longo do tempo, tornando-se um sistema legado. No segundo tema, serão descritos os componentes principais de um sistema legado, mostrando também uma organização estrutural em camadas próprias desses sistemas. O terceiro tema trata da avaliação de um sistema legado e descreve como tomar ações com relação à sua evolução. O quarto tema mostra como verificar o valor de negócio de um sistema legado, enquanto o quinto e último tema explica como averiguar tecnicamente um sistema legado.

CONTEXTUALIZANDO

Como visto na aula anterior, as leis de Lehman compreendem o processo de evolução de sistemas de larga escala. Apesar dessas leis não se aplicarem a sistemas de dispositivos móveis por serem programas menores, muitos dos conceitos apresentados anteriormente e dos problemas tratados nesta aula se aplicam, sim, ao desenvolvimento de sistemas em dispositivos móveis, assim como qualquer outro software. Um dos pontos principais discutidos na aula anterior, e que será discutido também nesta aula, é o cuidado com boas práticas de desenvolvimento de software para evitar a degradação da qualidade de um programa. Sem o devido cuidado, qualquer software tende a ter o seu código degradado. Lembre-se de que um sistema legado também pode ser interpretado como um sistema cujo código não apresenta testes automatizados para garantir a sua qualidade, e isso se aplica também a softwares de dispositivos móveis.

TEMA 1 – DEGRADAÇÃO DE SISTEMAS AO LONGO DO TEMPO

Necessidade de mudanças é o motivo pelo qual os sistemas evoluem. Tais mudanças podem provir de requisitos existentes não implementados na última versão liberada, de pedidos por novos requisitos, de relatórios de defeitos dos usuários ou de novas ideias de melhorias, vindas do time de desenvolvimento. Os processos de identificação de mudança e de evolução de sistema são cíclicos e continuam durante o ciclo de vida de um sistema.

Durante o processo de evolução, os requisitos são analisados em detalhes, de modo que as implicações das mudanças no sistema, que não foram

identificadas inicialmente, se tornam evidentes. As mudanças propostas devem ser modificadas e discutidas com os usuários antes de serem implementadas.

Pedidos de mudança, às vezes, estão relacionados com problemas no sistema, os quais devem ser resolvidos imediatamente. Tais mudanças urgentes podem surgir pelos seguintes motivos:

1. Se um problema sério no sistema deve ser corrigido para permitir a sua operação normal.
2. Se as mudanças aplicadas no ambiente em que o sistema é executado têm efeitos inesperados e afetam a sua operação normal.
3. Se existem mudanças identificadas tardiamente a serem feitas no negócio do sistema, como implementação de funcionalidades para competir com novos concorrentes, ou atender a alguma norma ou padrão de mercado, por exemplo.

Nesses casos, a necessidade de se fazer a mudança de forma rápida significa que não será possível seguir o processo formal de análise de mudanças. Em vez de modificar os requisitos e a arquitetura, a correção de emergência é feita para resolver o problema imediato. Como consequência, a documentação dos requisitos e da arquitetura do software fica inconsistente com o código. Mesmo que se tenha a intenção de documentar as alterações nos requisitos e na arquitetura posteriormente, novas correções adicionais de emergência podem ser necessárias. Tais correções terão prioridade sobre atualização de documentação. Com isso, a mudança original é esquecida e o código e a documentação deixam de estar alinhados.

Reparos de emergência geralmente têm de ser completados o mais rápido possível, e soluções rápidas e funcionais costumam ser escolhidas em detrimento das melhores soluções. Isso acelera o processo de envelhecimento, fazendo com que futuras mudanças sejam progressivamente mais difíceis, aumentando os custos de manutenção.

Idealmente, quando reparos de emergência são feitos, o pedido de mudança original deve permanecer em aberto até que os defeitos sejam corrigidos. Posteriormente, o pedido de mudança pode ser reimplementado de forma mais cuidadosa depois de uma análise mais aprofundada da solução e do sistema. O código da correção pode ser reutilizado, e uma solução alternativa e melhor para o problema pode ser feita, se houver mais tempo para a análise.

Na prática, todavia, é muito provável que essas melhorias terão baixa

prioridade para serem implementadas. Geralmente são esquecidas e, se mais mudanças são feitas ao longo do tempo, acaba se tornando inviável refazer as alterações de emergência.

Seguem outras causas da degradação do sistema ao longo do tempo:

1. **Instabilidade da equipe de desenvolvimento:** Depois que um sistema é liberado, é normal que a equipe de desenvolvimento seja dividida e que os integrantes trabalhem em outros projetos. O novo time (indivíduos responsáveis pela manutenção do sistema) não entende o sistema ou o motivo das decisões feitas para a sua arquitetura, e precisa, portanto, de tempo para compreender o sistema antes de implementar mudanças.
2. **Práticas falhas de desenvolvimento:** O contrato para manter um sistema é, geralmente, separado do contrato de desenvolvimento. O contrato de manutenção pode ser dado a uma empresa terceira. A presença desse fator, além da falta de estabilidade do time de desenvolvimento, significa que não existe incentivo para que o time terceiro desenvolva um software de fácil manutenção. Se o time de desenvolvimento terceiro pode implementar mudanças de forma mais rápida e, conseqüentemente, com menos esforço, isso compensará, mesmo que o código do sistema seja mais difícil de modificar no futuro.
3. **Habilidades do time:** O time de manutenção costuma ser relativamente inexperiente e novato com relação ao domínio de aplicação do sistema. O trabalho de manutenção tem uma imagem de segundo escalão entre engenheiros de software. É visto como um trabalho que exige menos habilidades, comparado ao trabalho de desenvolvimento, e é geralmente atribuído a um time mais inexperiente. Além disso, sistemas antigos costumam ser feitos em linguagens de programação obsoletas. O time de manutenção pode não ter muita experiência com tais linguagens, e deve aprendê-las para manter o sistema.
4. **Idade do programa e estrutura:** Assim que as mudanças forem feitas nos programas, suas estruturas tendem a degradar. Conseqüentemente, à medida que os programas envelhecem, eles se tornam mais difíceis de serem compreendidos e modificados. Muitos sistemas podem ter sido desenvolvidos sem técnicas modernas de engenharia de software. Podem nunca ter sido bem estruturados e podem, talvez, ter sido otimizados para serem eficientes, e não necessariamente para que sejam de fácil

compreensão. A documentação do sistema pode ter sido perdida, ou está inconsistente com o código do sistema. Sistemas antigos podem não ser controlados por processos de gerência de configuração, o que faz com que tempo considerável seja despendido para identificar as versões certas dos componentes para aplicar modificações.

Os três primeiros problemas acontecem porque muitas organizações consideram que as atividades de manutenção e de desenvolvimento devem ser separadas. Manutenção é vista como uma atividade de segunda classe, e não há incentivo para despendar dinheiro durante seu desenvolvimento para reduzir custos nas mudanças do sistema. A única solução a longo prazo para este problema é aceitar que sistemas raramente têm um tempo de vida definido, e que o seu uso continua, de alguma forma, por tempo indefinido. Sistemas evoluem durante o seu tempo de vida, seguindo um processo de desenvolvimento contínuo.

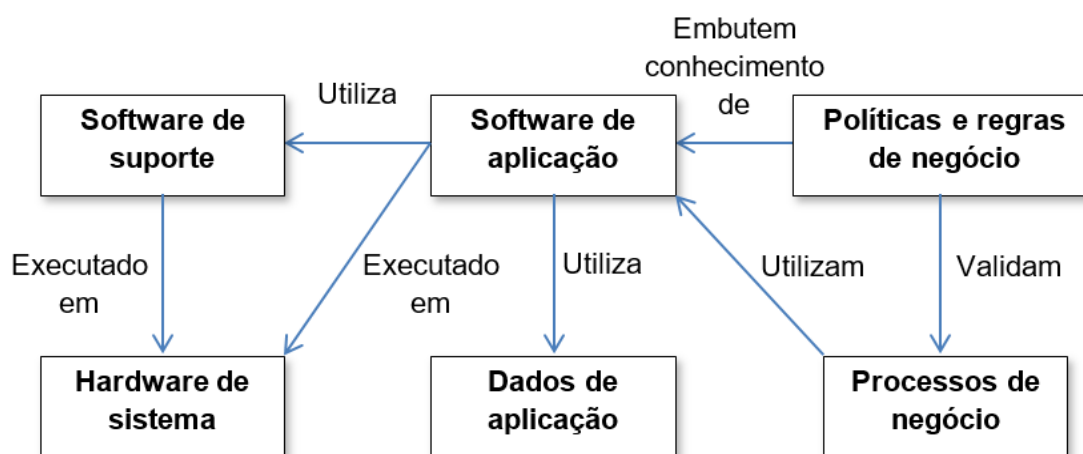
O quarto problema, degradação da estrutura do sistema, é o mais fácil de resolver, se comparado a todos os problemas anteriormente citados. Técnicas de engenharia de software, as quais serão abordadas em aulas posteriores, podem ser aplicadas para melhorar a estrutura do sistema, assim como o seu entendimento. Mudanças de arquitetura podem adaptar o sistema para novos equipamentos. A refatoração pode melhorar a qualidade do código do sistema, fazendo com que seja mais fácil de modificá-lo.

Neste tema, foram abordados os motivos principais da degradação de um sistema ao longo do tempo, tornando-o um sistema legado. No próximo tema, serão abordadas as leis de Lehman, que definem como funcionam as mudanças de sistemas computacionais.

TEMA 2 – COMPONENTES DE SISTEMAS LEGADOS

A Figura a seguir ilustra os componentes de sistemas legados:

Figura 1 – Componentes de sistemas legados



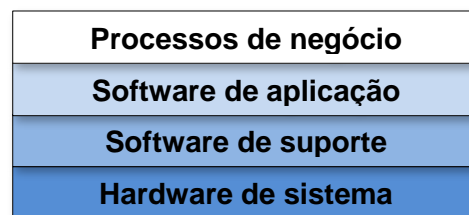
1. **Hardware do sistema:** sistemas legados podem ter sido feitos para hardwares não mais disponíveis, caros de manter e que podem não ser compatíveis com políticas organizacionais de TI em vigência para compra de novos equipamentos.
2. **Software de suporte (plataforma):** o sistema legado pode se basear em um conjunto de software de suporte, do sistema operacional e de utilidades fornecidas pelo fabricante do hardware a compiladores utilizados para o desenvolvimento do sistema. Este conjunto de software pode estar obsoleto e não ser mais suportado pelos seus mantenedores originais.
3. **Software da aplicação:** o sistema de aplicação que disponibiliza os serviços de negócio é, em geral, composto por vários programas separados desenvolvidos em momentos diferentes. Algumas vezes, o termo “sistema legado” se refere a este software, sem considerar todo o sistema.
4. **Dados da aplicação:** são os dados processados pelo sistema de aplicação. Em muitos sistemas legados, um volume imenso de dados foi acumulado durante o ciclo de vida do sistema. Tais dados podem estar inconsistentes e duplicados em vários arquivos.
5. **Processos de negócio:** são processos utilizados no negócio para atingir algum objetivo. Um exemplo de processo de negócio em uma companhia de seguros seria uma emissão de uma apólice de seguro; em uma indústria, poderia ser a aceitação de um pedido de produtos e a configuração do processo de fabricação destes produtos. Processos de

negócio podem ser idealizados com base em um sistema legado, sendo limitados por suas funcionalidades.

6. **Políticas e regras de negócio:** são definições sobre como o negócio deve ser executado, assim como suas regras. A utilização do sistema de aplicação pode estar embutida nestas políticas e regras de negócio.

Uma forma alternativa de se visualizar estes componentes é por uma série de camadas, como ilustrado na Figura abaixo.

Figura 2 – Camadas de um sistema legado



Fonte: O autor.

Cada camada depende da sua sucessora imediata, de modo que elas se inter-relacionem. Se o relacionamento entre tais camadas for mantido, deve ser possível efetuar mudanças em uma camada sem afetar as camadas próximas a ela. Todavia, este encapsulamento simples é uma simplificação exagerada, e mudanças em uma camada do sistema podem requerer outras mudanças em camadas localizadas acima e abaixo da camada afetada. Os motivos para isso são os seguintes:

- Mudanças em uma camada do sistema podem introduzir novas funcionalidades, e camadas mais altas do sistema podem ser modificadas para utilizá-las. Por exemplo, um novo banco de dados, introduzido na camada de suporte, pode compreender rotinas de acesso a dados pelo navegador de internet, e processos de negócio podem ser modificados para explorar os benefícios dessa mudança.
- Mudanças no software podem acrescentar uma carga extra de processamento no sistema, fazendo com que o hardware seja atualizado para melhorar seu desempenho. A melhoria de desempenho com o novo hardware pode fazer com que mudanças de software, antes impraticáveis, se tornem possíveis.

TEMA 3 – AVALIAÇÃO DE SISTEMAS LEGADOS

Muitas organizações têm um portfólio de sistemas legados em utilização, com um orçamento limitado para manter e atualizar tais sistemas. Estas organizações precisam decidir como obter o melhor retorno do seu investimento.

Isso envolve realizar uma análise realística dos seus sistemas para, então, decidir a estratégia mais apropriada para fazê-los evoluir. Em geral, existem quatro opções estratégicas:

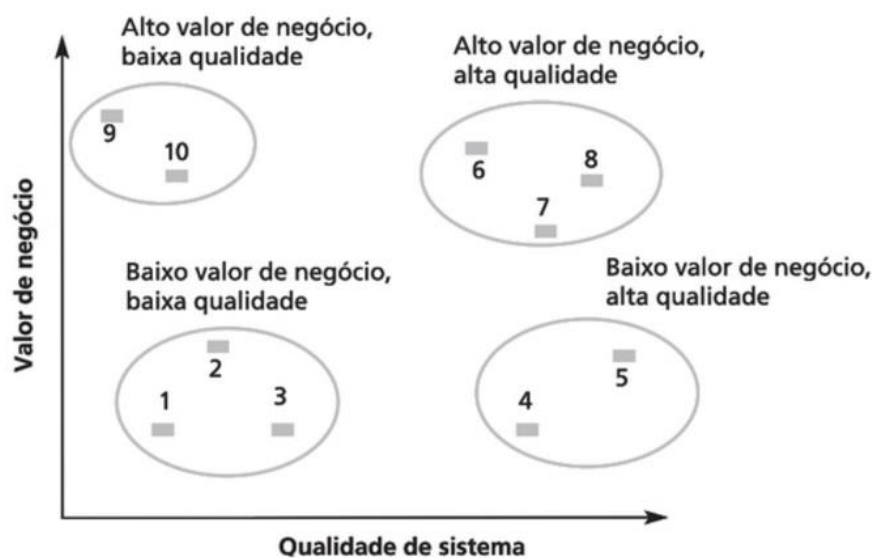
1. **Retirar o sistema por completo:** esta opção deve ser escolhida quando o sistema não está tendo uma utilização efetiva na contribuição para os processos de negócio. Isso ocorre com frequência quando processos de negócio sofreram modificações desde que o sistema foi instalado e que não dependem mais do sistema legado.
2. **Deixar o sistema sem modificações e continuar com a manutenção regular:** esta opção deve ser escolhida quando o sistema ainda é utilizado, possui operação estável e os seus usuários fazem poucas requisições de mudança.
3. **Efetuar reengenharia do sistema para melhorar a facilidade de manutenção:** esta opção deve ser escolhida quando a qualidade do sistema foi degradada e existem novas mudanças sendo propostas. Este processo pode incluir o desenvolvimento de novos componentes de interface para que o sistema original possa ser integrado a outros sistemas mais modernos.
4. **Substituir todo ou parte do sistema por um novo:** esta opção deve ser escolhida quando fatores de atualização de sistema, como aquisição de novo hardware, indicam que o sistema antigo não pode continuar, ou que existem sistemas de base que permitiriam que um novo sistema fosse desenvolvido a um custo razoável. Em muitos casos, uma estratégia de substituição evolucionária pode ser adotada, de modo que os componentes de sistema principais sejam substituídos por sistemas de base, junto de outros componentes reusáveis, quando possível.

Naturalmente, essas opções não são exclusivas. Quando um sistema é composto por vários programas, diferentes opções podem ser aplicadas para cada programa. Quando se está assessorando um sistema legado, deve-se avaliar o sistema sobre perspectivas de negócio e técnica. De uma perspectiva de

negócio, é preciso decidir se o negócio realmente precisa ou não do sistema. De uma perspectiva técnica, deve-se assessorar a qualidade do software de aplicação, o software de suporte e o hardware. Considerando estas perspectivas, a sugestão é utilizar uma combinação do valor de negócio com a qualidade do sistema para informar sua decisão.

Por exemplo, suponha que uma organização tenha dez sistemas legados. É preciso, portanto, assessorar a qualidade e o valor de negócio de cada um desses sistemas. Gera-se, então, gráfico ilustrando a relação entre o valor de negócio e a qualidade do sistema para análise, como mostra a Figura 3.

Figura 3 – Exemplo de uma assessoria em sistemas legados



Fonte: Sommerville (2011).

Na Figura 3, podem ser vistas quatro características de sistemas legados:

1. **Baixa qualidade, baixo valor de negócio:** manter estes sistemas em operação será caro, e o retorno para o negócio será pequeno. Estes sistemas devem ser desativados.
2. **Baixa qualidade, alto valor de negócio:** estes sistemas têm uma importante contribuição para o negócio, e não podem ser desativados. Porém, sua baixa qualidade significa que é muito caro mantê-los. Eles devem, então, passar por um processo de reengenharia para melhorar sua qualidade. Podem ser substituídos se um sistema de base estiver disponível.
3. **Alta qualidade, baixo valor de negócio:** estes sistemas não contribuem o esperado para o negócio, mas não é muito caro para mantê-los ativos.

Não compensa substituir tais sistemas. Logo, as atividades normais de manutenção podem ser continuadas se mudanças extensas não são requeridas, e se o hardware do sistema ainda é de mercado. Se um alto investimento em mudanças precisa ser feito, o sistema deve ser desativado.

4. **Alta qualidade, alto valor de negócio:** estes sistemas devem ser mantidos em atividade. Sua alta qualidade significa que não se deve investir em grandes mudanças, ou na substituição do sistema. A manutenção do sistema deve continuar.

TEMA 4 – AVALIAÇÃO DE VALOR DE NEGÓCIO DE SISTEMAS LEGADOS

Para analisar o valor de negócio, os interessados no sistema devem ser identificados, como os usuários finais e os seus gerentes, para que seja possível realizar várias perguntas sobre o sistema. Existem quatro questões básicas a se discutir:

1. **A utilização do sistema:** se são apenas utilizados ocasionalmente, ou por um grupo pequeno de pessoas, estes sistemas têm um baixo valor de negócio. É possível que um sistema legado tenha sido desenvolvido para suprir uma necessidade de negócio que pode ter mudado, ou que agora pode ser mantida de formas mais efetivas. Porém, deve-se ter cuidado com utilizações ocasionais, mas importantes, do sistema. Por exemplo, em uma universidade, um sistema de registro de estudantes pode ser utilizado apenas no início de cada ano letivo. Todavia, é um sistema essencial, com um alto valor de negócio.
2. **Os processos de negócio que são suportados:** quando um sistema é introduzido, processos de negócio são feitos para aproveitar a capacidade do sistema. Se o sistema é inflexível, a modificação de tais processos de negócio pode ser impossível. À medida que o ambiente se modifica, os processos de negócio originais podem se tornar obsoletos. Como consequência, um sistema pode ter um baixo valor de negócio porque força a utilização de processos de negócio ineficientes.
3. **A confiança do sistema:** a falta de confiança no sistema não é apenas um problema técnico, mas também um problema de negócio. Se um sistema não é confiável e os problemas afetam diretamente os clientes do negócio, ou ocorre que pessoas envolvidas no negócio se vejam obrigadas a parar

suas atividades para resolver esses problemas, o sistema tem um baixo valor de negócio.

4. **As saídas do sistema:** a questão chave, nesse caso, é a importância das saídas do sistema para o funcionamento efetivo do negócio. Se o negócio depende dessas saídas, então o sistema tem um alto valor de negócio. De forma alternativa, se essas saídas podem ser geradas de outra forma, ou se o sistema produz saídas raramente utilizadas, então o valor de negócio do sistema pode ser baixo.

TEMA 5 – AVALIAÇÃO TÉCNICA DE SISTEMAS LEGADOS

Para assessorar um sistema de uma perspectiva técnica, é necessário considerar o sistema de aplicação e o ambiente em que o sistema opera. O ambiente inclui o hardware e todos os softwares de apoio (compiladores, ambientes de desenvolvimento, sistema operacional etc.) requeridos para manter o sistema. O ambiente é importante, pois muitas mudanças a serem feitas no sistema têm o intuito de adequar mudanças já realizadas no ambiente, como atualizações de hardware ou de sistema operacional.

Tabela 1 – Fatores de avaliação técnica da plataforma legada

Fator	Questões
Estabilidade do fornecedor	O fornecedor ainda existe? O fornecedor é financeiramente estável e deve continuar existindo? Se o fornecedor não está mais no negócio, existe alguém que mantém os sistemas?
Taxa de falhas	O hardware tem uma grande taxa de falhas reportadas? O software de apoio trava e força o reinício do sistema?
Idade	Quantos anos têm o hardware e o software? Quanto mais velhos o hardware e o software de apoio, mais obsoletos serão. Ainda podem funcionar corretamente, mas poderia haver significativos benefícios econômicos e empresariais se migrassem para um sistema mais moderno.
Desempenho	O desempenho do sistema é adequado? Os problemas de desempenho têm um efeito significativo sobre os usuários do sistema?
Requisitos de apoio	Qual apoio local é requisitado pelo hardware e pelo software? Se houver altos custos associados a esse apoio, pode valer a pena considerar a substituição do sistema.
Custos de manutenção	Quais são os custos de manutenção de hardware e de licenças de software de apoio? Os hardwares mais antigos podem ter custos de manutenção mais elevados do que os sistemas modernos. Os softwares de apoio podem ter altos custos de licenciamento anual.
Interoperabilidade	Existem problemas de interface do sistema com outros sistemas? Compiladores podem, por exemplo, ser usados com as versões atuais do sistema operacional? É necessária a emulação do hardware?

É recomendável, durante o processo de assessoria, efetuar medições do sistema e dos seus processos de manutenção. Exemplos de dados que podem ser úteis incluem os custos de manutenção do hardware do sistema e do software de suporte (plataforma), o número de falhas no hardware que ocorrem durante um período de tempo e a frequência de aplicação de patches e correções no software de suporte do sistema.

Na Tabela 1, são mostrados alguns fatores que devem ser considerados durante a assessoria. Observe que nem todos consistem em características técnicas do ambiente. Deve-se considerar também a confiabilidade de fornecedores de hardware e de software de suporte. Se esses fornecedores não estiverem mais no mercado, pode não haver mais suporte para estes sistemas.

Tabela 2 – Fatores de avaliação técnica do sistema de aplicação legado

Fatores	Questões
Inteligibilidade	Quão difícil é compreender o código-fonte do sistema atual? Quão complexas são as estruturas de controle usadas? As variáveis têm nomes significativos que refletem a sua função?
Documentação	Qual documentação do sistema está disponível? A documentação é completa, consistente e atual?
Dados	Existe um modelo de dados explícito para o sistema? Até que ponto os dados nos arquivos estão duplicados? Os dados usados pelo sistema são atuais e consistentes?
Desempenho	O desempenho da aplicação é adequado? Os problemas de desempenho têm um efeito significativo sobre os usuários do sistema?
Linguagem de programação	Compiladores modernos estão disponíveis para a linguagem de programação usada para desenvolver o sistema? A linguagem de programação ainda é usada para o desenvolvimento do novo sistema?
Gerenciamento de configuração	Todas as versões de todas as partes do sistema são gerenciadas por um sistema de gerenciamento de configuração? Existe uma descrição explícita das versões de componentes usadas no sistema atual?
Dados de teste	Existem dados de teste para o sistema? Existem registros dos testes de regressão feitos quando novos recursos forem adicionados no sistema?
Habilidade de pessoal	Existem pessoas disponíveis com as habilidades necessárias para manter a aplicação? Existem pessoas disponíveis que tenham experiência no sistema?

Para assessorar a qualidade técnica de um sistema de aplicação, deve-se verificar uma variedade de fatores (Tabela 2) primariamente relacionados com a confiança do sistema, as dificuldades de manutenção do sistema e da sua documentação. Pode-se também coletar dados que ajudarão na análise da qualidade do sistema. Dados que podem ser úteis na análise são:

1. **Número de requisições de mudanças no sistema:** mudanças no sistema usualmente degradam a sua estrutura, fazendo com que mudanças futuras

sejam mais difíceis de serem implementadas. Quanto maior for este valor, menor pode ser a qualidade do sistema.

2. **O número de interfaces de usuário:** este é um fator importante em sistemas que apresentam interação com usuários, de modo que cada interface, nesse caso, é considerada como uma tela ou um formulário. Quanto maior o número de interfaces, maior a probabilidade de que haverá inconsistências e redundâncias entre elas.
3. **O volume de dados utilizado pelo sistema:** quanto maior for o volume de dados (número de arquivos, tamanho do banco de dados etc.), maior a probabilidade de que existam inconsistências que reduzem a qualidade do sistema.

FINALIZANDO

Nesta aula, discutimos a degradação de sistemas legados, enfatizando os problemas que ocorrem em consequência de alterações urgentes feitas nos sistemas, sem investimento na sua qualidade ao longo do tempo. Depois, detalhamos a estrutura de sistemas legados, descrevendo seus componentes e sua organização em camadas. Após este tema, explicamos como realizar uma avaliação de sistemas legados, analisando seu valor de negócio para organização, assim como a sua qualidade técnica. Com base nestes fatores, tornou-se possível estabelecer uma estratégia de ação em relação à sua evolução futura. No tema seguinte, discutimos sobre o processo de avaliação do valor de negócio de um sistema legado. Por último, entendemos o modo como se deve avaliar um sistema tecnicamente, efetuando uma análise da plataforma e da aplicação do sistema legado.

REFERÊNCIAS

SOMMERVILLE, I. **Engenharia de software**. Trad. Kalinka Oliveira e Ivan Bosnic. 9 ed. São Paulo: Pearson Prentice Hall, 2011.

SISTEMAS legados. Disponível em: <<https://ifs.host.cs.st-andrews.ac.uk/Books/SE9/Web/LegacySys/>>. Acesso em: 18 set. 2017.