Por que fazer a modelagem?

Uma empresa de software bem-sucedida é aquela que fornece software de qualidade e capaz de atender às necessidades dos respectivos usuários. Uma empresa que consiga desenvolver esse software de maneira previsível e em determinado período, com utilização eficiente e eficaz de recursos, será uma empresa com um negócio viável.

Existe uma implicação importante nessa mensagem: o principal produto de uma equipe de desenvolvimento não são documentos bonitos, reuniões sofisticadas, ótimos slogans ou linhas de código-fonte merecedoras do Prêmio Pulitzer. O principal produto é um bom software capaz de satisfazer às necessidades de seus usuários e respectivos negócios. Tudo o mais é secundário.

Infelizmente, muitas empresas de software confundem “secundário” com “irrelevante”. Para entregar um software que satisfaça ao propósito pretendido, será preciso reunir-se e interagir com os usuários de uma maneira disciplinada, com a finalidade de expor os requisitos reais do sistema. Para desenvolver software de qualidade duradoura, será necessário criar uma arquitetura de fundação sólida que aceite modificações. Para desenvolver software de forma rápida, eficiente e efetiva, com o mínimo de desperdício e de retrabalho de software, será preciso dispor das pessoas certas, das ferramentas adequadas e do enfoque correto. Para fazer tudo isso de maneira previsível e consistente, com uma avaliação dos custos reais do sistema, você precisará de um processo seguro de desenvolvimento que possa ser adaptado às novas necessidades de seu negócio e de sua tecnologia.

A modelagem é uma parte central de todas as atividades que levam à implantação de um bom software. Construímos modelos para comunicar a estrutura e o comportamento desejados do sistema. Construímos modelos para visualizar e controlar a arquitetura do sistema. Construímos modelos para compreender melhor o sistema que estamos elaborando, muitas vezes expondo oportunidades de simplificação e reaproveitamento. Construímos modelos para gerenciar os riscos.

A importância da modelagem

Para construir uma casa para seu cachorro, você poderá começar juntando uma pilha de tábuas, alguns pregos e algumas ferramentas básicas, como martelo, serrote e metro. Em poucas horas, com pequeno planejamento prévio, provavelmente a casa de cachorro estará pronta, razoavelmente funcional e certamente você será capaz de fazer tudo isso sem precisar da ajuda de ninguém. Desde que a nova casa seja suficientemente grande e não haja muitas goteiras, seu cão ficará feliz. Se não der certo, sempre será possível fazer tudo de novo ou arrumar um cachorro menos exigente.

Para construir uma casa para sua família, você poderá começar juntando uma pilha de tábuas, alguns pregos e algumas ferramentas básicas, mas precisará de um tempo muito maior e, com certeza, sua família será mais exigente que o cachorro. Nesse caso, a menos que já tenha construído várias casas, será melhor fazer um planejamento detalhado antes de prender o primeiro prego ou iniciar a fundação. Pelo menos, você fará alguns desenhos rápidos da aparência desejada para sua futura casa. Se quiser construir uma casa de qualidade, que atenda às necessidades de sua família e respeite os códigos de edificação da região, também será preciso desenhar alguns esboços de projetos, com a finalidade de pensar sobre o uso pretendido para cada cômodo e detalhes práticos de energia, circulação e encanamento. A partir desses planos, você poderá começar a fazer uma estimativa razoável da quantidade de tempo e de material necessários para essa tarefa. Embora seja humanamente possível construir uma casa sozinho, você logo descobrirá que será muito mais eficiente trabalhar com outras pessoas, talvez terceirizando vários serviços básicos ou comprando material pré-fabricado. Desde que você se mantenha fiel aos planos e permaneça dentro dos limites de tempo e custos, provavelmente sua família ficará satisfeita. Se não der certo, a solução não será trocar de família. Portanto, será melhor definer as expectativas desde o início e gerenciar qualquer modificação com muita cautela.

Para construir um prédio comercial com vários andares, não será uma boa ideia começar com uma pilha de tábuas, alguns pregos e algumas ferramentas básicas. Como provavelmente você estará usando o dinheiro de outras pessoas, como os acionistas da empresa, elas exigirão saber

o tamanho, a forma e o estilo do futuro prédio. Muitas vezes, essas pessoas mudarão de ideia, mesmo depois de iniciada a construção. Valerá a pena fazer um planejamento rigoroso, pois os custos de qualquer erro serão altos. Você será apenas uma parte de um grupo bem maior, responsável pelo desenvolvimento e pela entregado prédio. Assim, a equipe precisará de todos os modelos e esboços do projeto para poderem se comunicar entre si. Desde que você consiga as pessoas certas e as ferramentas adequadas, além de gerenciar de maneira ativa o processo de transformação de um conceito de arquitetura em realidade, provavelmente acabará obtendo um prédio que satisfará seus futuros ocupantes. Caso pretenda continuar construindo prédios, você tentará encontrar um equilíbrio entre os desejos dos futuros ocupantes e a realidade das tecnologias de construção, além de manter um relacionamento profissional com os demais integrantes de sua equipe, nunca colocando-os em risco, nem exigindo tanto que eles acabem exaustos ou doentes.

Curiosamente, muitas empresas de desenvolvimento de software começam querendo construir prédios altos, como se estivessem fazendo uma casinha de cachorro.

De vez em quando, a sorte ajuda. Com as pessoas certas no momento adequado e com todos os planetas em um alinhamento favorável, talvez, e apenas talvez, você consiga fazer com que sua equipe crie um produto de software capaz de encantar seus usuários. Tipicamente, entretanto, você não conseguirá todas as pessoas certas (já que estarão superocupadas), nunca será o momento adequado (ontem teria sido muito melhor) e dificilmente os planetas estarão alinhados (movendo-se de uma maneira que você não poderá modificar). Considerando-se a demanda cada vez maior de desenvolvimento de software nesta época de Internet, as equipes de desenvolvimento costumam se restringir à tarefa que realmente sabem fazer bem: digitar linhas de código. Esforços heróicos de programação já viraram lendas nessa indústria e, de uma forma geral, tem-se a impressão de que apenas trabalhar bastante seja a resposta adequada a qualquer crise que ocorra no processo de desenvolvimento. Porém, esse trabalho não produzirá necessariamente as linhas de código corretas, além de alguns projetos serem de tal grandeza que até o acréscimo de mais horas de trabalho não será suficiente para a conclusão da tarefa.

Se você realmente quiser construir softwares equivalentes a uma casa ou a um prédio, o problema não se restringirá a uma questão de escrever uma grande quantidade de software, de fato, o segredo estará em criar o código correto e pensarem como será possível elaborar menos software. Isso faz com que o desenvolvimento de software de qualidade se torne uma questão de arquitetura, processo e ferramentas. Ainda assim, muitos projetos são iniciados parecendo uma casa de cachorro, mas crescem com a grandeza de um prédio simplesmente porque são vítimas de seu próprio sucesso. Chega um momento em que, caso não tenham sido consideradas questões referentes à arquitetura, a processos e a ferramentas, a casa de cachorro, agora ampliada para um grande prédio, sucumbirá ao seu próprio peso. Qualquer erro na casa de cachorro poderá deixar seu cão insatisfeito. A falha na construção de um grande prédio afetará materialmente seus ocupantes.

Projetos de software mal sucedidos falham em relação a aspectos únicos e específicos de cada projeto, mas todos os projetos bem-sucedidos são semelhantes em diversos aspectos. Existem muitos elementos que contribuem para uma empresa de software de sucesso; um desses componentes é a utilização da modelagem.

A modelagem é uma técnica de engenharia aprovada e bem-aceita. Construímos modelos de arquitetura de casas e de grandes prédios para auxiliar seus usuários a visualizar qual será o produto final. Podemos até elaborar modelos matemáticos coma finalidade de analisar quais serão os efeitos que ventos e tremores de terra poderão causar aos prédios que você construir.

A modelagem não faz parte apenas da indústria de construção. Seria inconcebível fornecer um novo avião ou automóvel sem primeiro construiros respectivos modelos – desde modelos de computadores, modelos físicos de túneis de vento até protótipos em larga escala. Novos dispositivos elétricos, desde microprocessadores a sistemas de telefonia, requerem algum grau de modelagem com o propósito de permitir uma melhor compreensão do sistema e a comunicação dessas ideias a outras pessoas. Na indústria cinematográfica, a elaboração de roteiros, que é uma forma de modelagem, tem um papel central em qualquer produção. Nas áreas de sociologia, economia e administração comercial, construímos modelos para validar nossas teorias, ou experimentar novas, com riscos e custo-mínimos.

Afinal, o que é um modelo? Falando de uma maneira simples:

Um modelo é uma simplificação da realidade.

Os modelos fornecem uma cópia do projeto de um sistema. Os modelos poderão abranger planos detalhados, assim como planos mais gerais com uma visão panorâmica do sistema considerado. Um bom modelo inclui aqueles componentes que têm ampla repercussão e omite os componentes menores que não são relevantes em determinado nível de abstração. Todos os sistemas podem ser descritos sob diferentes aspectos, com a utilização de modelos distintos, e cada modelo será, portanto, uma abstração semanticamente especifica do sistema. Os modelos podem ser estruturais, dando ênfase à organização do sistema, ou podem ser comportamentais, dando ênfase à dinâmica do sistema.

Por que fazera modelagem? Existe um motivo fundamental.

Construímos modelos para compreender melhor o sistema que estamos desenvolvendo.

Com a modelagem, alcançamos quatro objetivos.

1. Os modelos ajudam a visualizar o sistema como ele é ou como desejamos que seja.
2. Os modelos permitem especificar a estrutura ou o comportamento de um sistema.
3. Os modelos proporcionam um guia para a construção do sistema.
4. Os modelos documentam as decisões tomadas.

A modelagem não se restringe a grandes sistemas. Até os softwares equivalentes a casas de cachorro poderão receber os benefícios da modelagem. Porém, é absolutamente verdadeiro que, quanto maior e mais complexo for o sistema, maior será a importância da modelagem, por uma razão muito simples:

Construímos modelos de sistemas complexos porque não é possível compreendê-los em sua totalidade.

Existem limites para a capacidade humana de compreender complexidades. Com a ajuda da modelagem, delimitamos o problema que estamos estudando, restringindo nosso foco a um único aspecto por vez. Em essência esse é o procedimento de "dividir-e-conquistar", do qual Edsger Dijkstra falava há anos: ataque um problema difícil, dividindo-o em vários problemas menores que você pode solucionar. Além disso, com o auxílio da modelagem, somos capazes de ampliar o intelecto humano. Um modelo escolhido de maneira adequada permitirá a quem usa a modelagem trabalhar em níveis mais altos de abstração.

Afirmar que a modelagem deveria ser feita não significa necessariamente que ela será utilizada. De fato, vários estudos sugerem que, em sua maioria, as empresas de software usam pouca, quando usam alguma modelagem formal. Compare o uso da modelagem com a complexidade de um projeto e você descobrirá que, quanto mais simples for o projeto, menos provável será a utilização da modelagem formal.

A palavra funcional aqui utilizada é "formal". Na verdade, inclusive nos projetos mais simples, os desenvolvedores realizam alguma modelagem, apesar da maneira muito informal. O desenvolvedor poderá rabiscar uma ideia em um quadro-negro ou em uma folha de papel, com a finalidade de visualizar parte do sistema, ou a equipe poderá usar cartões CRC para trabalharem um cenário específico ou estruturar determinado mecanismo. Não há nada errado nesses modelos. Se funcionarem, sem dúvida deverão ser utilizados. Porém, esses modelos informais costumam ser ad hoc e não oferecem uma linguagem básica que possa ser compartilhada com outras pessoas facilmente. Assim como existe uma linguagem básica para esboços de projetos na indústria de construção, na engenharia elétrica e na modelagem matemática, também uma empresa de desenvolvimento de software poderá usufruir os benefícios de utilizar uma linguagem básica para a modelagem de software.

Qualquer projeto será beneficiado pelo uso de algum tipo de modelagem, Inclusive no setor de softwares comerciais, em que às vezes é mais comum distribuir softwares inadequados devido à produtividade oferecida pelas linguagens de programação visual, a modelagem poderá auxiliar a equipe de desenvolvimento a visualizar melhor o planejamento do sistema e permitir que o desenvolvimento seja mais rápido, ajudando a construir o item correto. Quanto mais

complexo for o sistema, maior será a probabilidade de ocorrência de erros ou de construção de itens errados, caso não haja qualquer modelagem. Todos os sistemas úteis e interessantes apresentam uma tendência natural para se transformarem em algo mais complexo ao longo do tempo. Portanto, ainda que considere não ser preciso fazer a modelagem hoje, à medida que o seu sistema evoluir, você se arrependerá dessa decisão, quando for tarde demais.

Princípios da modelagem

O uso da modelagem tem uma rica história em todas as disciplinas de engenharia. Essa experiência sugere quatro princípios básicos de modelagem. Em primeiro lugar:

A escolha dos modelos a serem criados tem profunda influência sobre a maneira como um determinado problema é atacado e como uma solução é definida.

Em outras palavras, escolha bem os seus modelos. Os modelos corretos iluminarão de modo brilhante os problemas de desenvolvimento mais complicados, proporcionando conclusões que simplesmente não seriam possíveis de outra maneira; modelos inadequados causarão confusões, desviando a atenção para questões irrelevantes.

Deixando de lado o desenvolvimento de software por um instante, suponha que você esteja tentando solucionar um problema de física quântica. Certos problemas, como a interação de fótons no tempo-espaço, implicam uma matemática maravilhosamente complexa. Escolha um modelo que não seja o de cálculo e imediatamente essa complexidade inerente se tornará manipulável. Nesse campo, esse é precisamente o valor dos diagramas de Feynmann, capazes de oferecer representações gráficas de problemas bastante complexos. De modo semelhante, em um domínio inteiramente diferente, suponha que você está construindo um novo prédio e está interessado em saber como ele se comportará quando houver fortes ventanias. Construindo um modelo físico e submetendo-o a testes de túneis de vento, você aprenderá algumas coisas interessantes, apesar de os materiais em escalas menores não se flexionarem exatamente como em escalas maiores. Assim, se elaborar um modelo matemático e depois submetê-lo a simulações, você aprenderá algumas coisas diferentes e provavelmente também

será capaz de trabalhar com um número maior de novos cenários do que se estivesse utilizando modelos físicos. A partir de testes contínuos e rigorosos com seus modelos, você acabará obtendo um nível de confiança muito superior em relação ao fato de que o sistema, cuja modelagem foi realizada, de fato se comportará da maneira esperada no mundo real.

Em relação aos softwares, a escolha de modelos poderá ser modificada, de maneira significativa, de acordo com sua visão de mundo. Construindo um sistema a partir da perspectiva de um desenvolvedor de banco de dados, provavelmente você atribuirá o foco a modelos de relacionamentos entre entidades, cujo comportamento tende a privilegiar *Stored Procedures* e os eventos que os iniciam. Construindo um sistema a partir da perspectiva de um analista de análise estruturada, provavelmente usara modelos centrados em algoritmos, com o respectivo fluxo de dados de um processo para outro. Construindo um sistema a partir da perspectiva de um desenvolvedor orientado a objetos, provavelmente trabalhará com um sistema cuja arquitetura estará centrada em várias classes e os padrões de interação que determinarão como essas classes funcionarão em conjunto. Qualquer uma dessas soluções poderá ser correta para uma determinada aplicação e cultura de desenvolvimento, apesar de a experiência indicar que a perspectiva orientada a objetos é superior para a criação de arquireturas flexíveis, inclusive no caso de sistemas que poderão conter grandes bancos de dados ou vários componentes computacionais. Apesar desse fato, o ponto mais importante é que cada visão de mundo conduz a um tipo diferente de sistema, com custos e benefícios diversos.

Em segundo lugar:

Cada modelo poderá ser expresso em diferentes níveis de precisão.

Ao desenvolver um sistema complexo, às vezes poderá ser necessária uma visão panorâmica – por exemplo, para ajudar os investidores a visualizar a aparência e o funcionamento do futuro sistema. Em outras situações, poderá ser preciso retornar ao nível dos alicerces – por exemplo, quando existe uma rotina cuja execução é crítica ou um componente estrutural pouco comum.

O mesmo é verdadeiro em relação aos modelos de software. Às vezes, um modelo da interface para o usuário, de execução rápida e simples, será exatamente o que você precisará; em outros casos, será necessário retornar a níveis mais baixos, como ao especificar interfaces para várias plataformas ou ao enfrentar congestionamentos em uma rede. Em qualquer situação, os melhores tipos de modelos serão aqueles que permitem a escolha do grau de detalhamento, dependendo de quem esteja fazendo a visualização e por que deseja fazê-la. Um analista ou um usuário final dirigirá a atenção em direção a questões referentes ao que será visualizado; o desenvolvedor moverá o foco para a maneira como esses objetos funcionarão. Todos esses observadores desejarão visualizar o sistema em níveis diferentes de detalhamento em situações distintas.

Em terceiro lugar:

Os melhores modelos estão relacionados à realidade.

O modelo físico de um prédio, que não responda da mesma forma que os materiais reais, terá um valor apenas limitado; o modelo matemático de um avião, em que são consideradas apenas condições de vôo ideais e fabricação perfeita, poderá ocultar características potencialmente fatais do avião de verdade. Será melhor utilizar modelos que tenham uma clara conexão com a realidade e, nos casos em que essa conexão seja fraca, saber, de maneira exata, como esses modelos diferem do mundo real. Todos os modelos simplificam a realidade; o segredo será ter certeza de que sua simplificação não ocultará detalhes importantes.

No caso dos softwares, o tendão-de-aquiles das técnicas de análise estruturada está no fato de não haver uma conexão básica entre o modelo de análise e o modelo de projeto do sistema. Falhando a ponte sobre essa fenda, ao longo do tempo aparecerá uma divergência entre o sistema concebido e o sistema construído. Nos sistemas orientados a objetos, é possível estabelecer alguma conexão entre todos os pontos de vista quase independentes de um sistema em uma mesma totalidade semântica.

Em quarto lugar:

Nenhum modelo único é suficiente. Qualquer sistema não-trivial será melhor investigado por meio de um pequeno conjunto de modelos quase independentes com vários pontos de vista.

Para a construção de um prédio, não existe um único conjunto de esboços de projetos capaz de revelar todos os detalhes da construção. Pelo menos, serão necessárias plantas baixas, aéreas, elétricas, de circulação e de água e esgoto.

A expressão funcional aqui utilizada é "quase independente". Nesse contexto, a expressão significa modelos que possam ser criados e estudados separadamente, mas que continuam inter-relacionados. Assim como no caso de um prédio, você poderá estudar apenas as plantas relacionadas à energia elétrica, mas também poderá ver o respectivo mapa na planta baixa e talvez até sua interação com o direcionamento de canos na planta de água e esgoto.

O mesmo é verdadeiro em relação a sistemas de software orientados a objetos. Para compreender a arquitetura desses sistemas, você precisará recorrer a várias visões complementares e inter-relacionadas: a visão dos casos de uso (expondo os requisitos do sistema), a visão de projeto (captando o vocabulário do espaço do problema e do espaço da solução), a visão do processo (modelando a distribuição dos processos e threads do sistema), a visão da implementação do sistema (direcionando a realização física do sistema) e a visão da implantação (como foco voltado para questões de engenharia de sistemas). Cada uma dessas visões poderá conter aspectos estruturais como também aspectos comportamentais. Em conjunto, essas visões representam a base do projeto do software.

Dependendo da natureza do sistema, alguns modelos poderão ser mais importantes do que outros. Por exemplo, no caso de sistemas que utilizam muitos dados, predominarão modelos voltados para a visão estática do projeto. Em sistemas centrados no uso de GUI, são importantes as visões estáticas e dinâmicas dos casos de uso. Em sistemas de execução crítica em tempo real, a visão dinâmica do processo tende a ser mais importante. Por fim, em sistemas distribuídos, como aqueles encontrados em aplicações que utilizam a Web, os modelos de implementação e de implantação são os mais importantes.

Referências Bibliográficas

BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. UML: Guia do Usuário. Rio de Janeiro: Elsevier, 2005.