LABORATÓRIO DE ENGENHARIA DE SOFTWARE

Gerenciamento de Requisitos de Software

Leffingwell, Dean & Widrig, Don. Managing Software Requirements: A Unified Approach – Addison-Wesley object technology series, Addison Wesley, 2000. ISBN: 0-201-61593-2

Tradução e Revisão Técnica

© Osvaldo Kotaro Takai,
e-mail: otakai@uol.com.br

Capítulo 15

Prototipação

Pontos chaves

- A prototipação é especialmente efetiva para atacar as síndromes "Sim, mas" e "Ruínas Desconhecidas".
- Um protótipo de requisitos de software é uma implementação parcial do sistema de software, construído para auxiliar os desenvolvedores, usuários e clientes a melhor entenderem os requisitos do sistema.
- Crie protótipos para requisitos "confusos": aqueles que, embora conhecidos ou implícitos, suas definições são pobres e mal compreendidas.

Os protótipos de software, como encarnações de um sistema de software, demonstram uma porção da funcionalidade de um novo sistema. Dado o que discutimos até aqui, nós esperamos que fique obvio de que a prototipação pode ser muito útil para descobrir necessidades do usuário. Usuários podem tocar, sentir e interagir com o protótipo do sistema de maneira que nenhuma das outras técnicas pode fornecer. De fato, a prototipação pode ser extremamente efetivo para atacar tanto a síndrome "Sim, mas" ("Isso não é exatamente o que eu queria"), quanto a síndrome das "Ruínas Desconhecidas" ("Agora que eu vi, eu tenho um outro requisito a adicionar").

Tipos de Protótipos

Protótipos podem ser categorizados de várias maneiras. Por exemplo, Davis (1995a) categoriza os protótipos como descartável versus evolucionário versus operacional, vertical versus horizontal, interface de usuário versus algorítmico, entre outras. O tipo de protótipo que você escolhe depende do problema que você está tentando resolver através da construção do protótipo.

Por exemplo, se o risco do seu projeto é baseado principalmente na viabilidade da abordagem tecnológica – isso, simplesmente nunca foi feito antes e você não está certo se a tecnologia aplicada pode atingir as metas de desempenho ou de rendimento – você pode querer desenvolver um *protótipo arquitetural* que principalmente demonstre a viabilidade da tecnologia a ser usada. Um protótipo arquitetural pode ainda ser *descartável* versus *evolucionário*. "Descartável" implica que o propósito do esforço é apenas para provar a viabilidade; assim, você poderá usar qualquer atalho, técnicas alternativas, simulações, ou qualquer outra coisa para atingir esse propósito. Quando você tiver terminado, você simplesmente o descarta, mantendo apenas o conhecimento apreendido nesse exercício. "Evolucionário" implica que você implementou o protótipo na mesma arquitetura que você pretende usar no sistema final, e que você será capaz de construir o sistema final a partir da evolução do protótipo.

Se a principal área de risco de seu projeto é a interface do usuário, por contrate, você irá querer desenvolver um *protótipo de requisitos*, usando qualquer tecnologia que permita a você, desenvolver interfaces do usuário muito rapidamente. A Figura 15–1 ilustra uma árvore de decisão que você pode usar para selecionar um tipo de protótipo que faça mais sentido para o seu projeto.

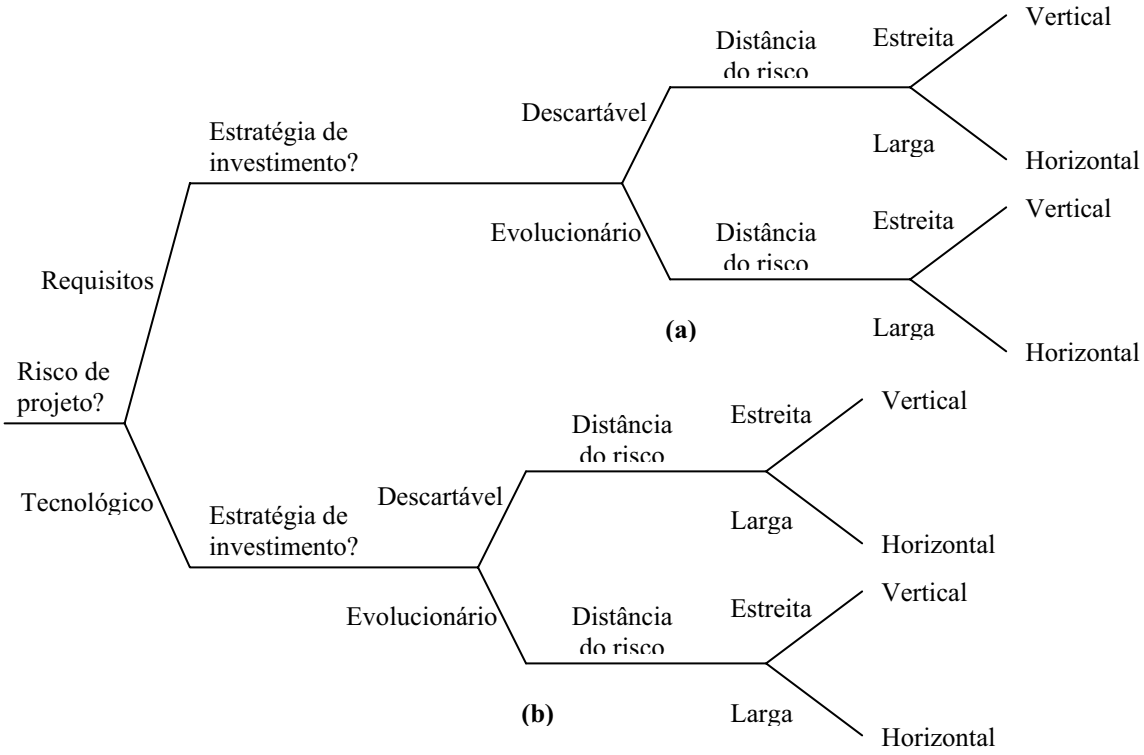

Figura 15–1 Árvore de decisão para seleção do tipo de protótipo: (a) protótipos de requisitos; (b) protótipos arquiteturais

Protótipos de Requisitos

Para propósitos de elucidação de requisitos, nós nos concentramos sobre os tipos de protótipos sob o ramo superior desta árvore. Definimos um protótipo de requisitos de software como:

> *uma implementação de um sistema de software, construída para ajudar desenvolvedores, usuários e clientes a melhor entender os requisitos do sistema.*

Com o propósito de elucidar requisitos, nós freqüentemente escolhemos construir um protótipo "descartável, horizontal, interface do usuário". "Horizontal" implica que nós iremos tentar construir uma grande quantidade de funcionalidade do sistema; um protótipo vertical, por outro lado, constroem-se apenas alguns requisitos de maneira qualitativa. "Interface de usuário" implica que nós iremos construir principalmente a interface do sistema para seus usuários ao invés de implementar a lógica e os algoritmos que residem dentro do software ou

prototipar interfaces para outros dispositivos ou sistemas. Como uma ferramenta de elucidação, os protótipos:

- Construídos por desenvolvedores, podem ser usados para obter confirmação de que o desenvolvedor entendeu os requisitos.
- Construídos por desenvolvedores, podem ser usados como um catalisador para encorajar o cliente a pensar em mais outros requisitos.
- Construídos pelo cliente, podem comunicar requisitos ao desenvolvedor.

Em todos os três casos, a meta é construir o protótipo de maneira que consuma poucos recursos. Se ficar muito caro construir, pode ser melhor construir o sistema real!

Muitos protótipos de software tendem a ser protótipos de requisitos e são usados principalmente para capturar aspectos da interface do usuário do sistema a ser construído. Existem provavelmente duas razões para isso:

1. A emergência de uma pletora[4] de ferramentas baratas e amplamente disponíveis para construir interfaces de usuários rapidamente.
2. Para sistemas cujas interfaces de usuário sejam intensas, um protótipo de interface de usuário revela, também, muitos outros requisitos, tais como as funções que são fornecidas ao usuário, quando cada função está disponível aos usuários e quais funções não são acessíveis aos usuários.

No entanto, precisamos estar certos de que a disponibilidade de ferramentas não nos leve a prototipar partes do sistema que não apresentem, inicialmente, riscos muito altos.

O que Prototipar

Como vamos saber qual porção do sistema devemos prototipar? Numa situação típica, nosso entendimento das necessidades dos usuários irá variar desde o bem conhecido e fácil de verbalizar até o totalmente desconhecido (Figura 15–2).

Figura 15–2 O sistema de estoque inicial, com atores identificados

Os requisitos bem-conhecidos podem ser óbvios no contexto do domínio da aplicação e a experiência do usuário e da equipe com sistemas desse tipo. Por exemplo, se estivermos simplesmente estendendo um sistema existente, teremos claramente a idéia de como a maioria das necessidades por novas funcionalidades deve ser. Os requisitos bem-conhecidos e bem-entendidos não precisam ser prototipados, a menos que sejam necessários para ajudar a visualizar o contexto de outras necessidades de usuários; construí-los irá consumir os já parcos recursos

[4] Superabundância qualquer, que produz efeito nocivo.

existentes, e como já estão bem-compreendidos, aprenderemos muito pouco com eles.

Os requisitos desconhecidos, no entanto, são as "Ruínas Desconhecidas" que nós desejamos conhecer. Infelizmente, nós não podemos realmente prototipá-los, se pudéssemos, não seria desconhecido! Assim, sobra como alvo de prototipação as partes "Confusas" que estão no meio. Esses requisitos podem ser conhecidos ou implícitos, mas sua definição e entendimento são pobres.

Construindo o Protótipo

A escolha da tecnologia usada para construção do protótipo depende das decisões tomadas considerando a árvore de decisão da Figura 15–1. Por exemplo, a escolha por um protótipo descartável da GUI nos leva a escolher qualquer tecnologia que seja barata e rápida para implementar exemplos de GUIs.

Se um protótipo evolucionário for selecionado, você deve escolher a linguagem e o ambiente de desenvolvimento que será utilizado para produzir a implementação final. Você também terá que fazer esforços significativos para projetar a arquitetura de software do sistema, bem como aplicar quaisquer padrões de codificação ou processos de software que você usará para criar o sistema. Caso contrário você terá que evoluir um sistema que fundamentalmente falha em um ou mais desses aspetos. Neste caso, você pode ter criado um protótipo descartável por acidente! Ou, pior, a qualidade do sistema implantado estará, para sempre, comprometida pelo seu protótipo de requisitos bem-intencionado.

Avaliando os Resultados

Depois que o protótipo estiver construído, ele deve ser exercitado pelos usuários num ambiente que simule, tanto quanto possível, o ambiente de produção no qual o sistema final será usado. Dessa forma, ambientes e outros fatores externos que afetam os requisitos do sistema também se tornarão óbvios. Além disso, é importante que existam vários tipos de usuários exercitem o protótipo, caso contrário os resultados serão preconceituosos.

Os resultados do processo de prototipação dividem-se em duas partes:

1. Necessidades confusas tornam-se melhor entendidas.
2. Ao exercitar o protótipo, inevitavelmente elucida respostas "Sim, mas" do usuário; assim, necessidades anteriormente desconhecidas tornam-se conhecidas. Simplesmente por enxergarem um conjunto de comportamentos, os usuários passam a entender outros requisitos que devem ser impostos ao sistema.

De qualquer forma, a prototipação virtualmente sempre produz resultados. Assim, você deve normalmente prototipar qualquer aplicação nova ou inovadora. O truque é assegurar que o retorno obtido no conhecimento dos requisitos faça valer o investimento realizado. Isso porque queremos, freqüentemente, prototipar – ou ao menos implementar nossos primeiros protótipos – rapidamente, utilizando técnicas baratas e disponíveis. Ao limitar o investimento, nós maximizamos o retorno sobre o investimento de obter o entendimento dos requisitos.

Sumário

Devido aos protótipos de software demonstrarem uma parte da funcionalidade desejada de um novo sistema, eles podem ser ferramentas efetivas para ajudar a refinar os requisitos reais do sistema. Eles são efetivos porque usuários podem *interagir* com um protótipo em seu ambiente, o qual é tão próximo ao mundo real quanto se puder chegar sem desenvolver o software de produção.

Você deve selecionar sua técnica de prototipação com base na probabilidade de um tipo de risco estar presente em seu sistema. Supõe-se que os protótipos de requisitos sejam baratos e fáceis de desenvolver, e que eles ajudem a eliminar grande parte dos riscos de requisitos de seu projeto.

Entre as várias possibilidades de investimento, você deve investir somente o necessário em seu protótipo. O uso de várias técnicas de prototipação, ou melhor, o uso combinado de diversas de técnicas de prototipação, tem se mostrado extremamente efetivo em auxiliar a equipe de projeto a desenvolver um melhor entendimento das reais necessidades de um sistema de software.