

Verificação, Validação e Priorização de Requisitos

Higor Dantas¹[118110808], Ivyna Alves²[000000000], Matheus Henrique³[118111780] e
Thaís Toscano⁴[000000000]

¹ Universidade Federal de Campina Grande, Campina Grande - PB, Brasil

`higor.dantas@ccc.ufcg.edu.br`

² `ivyna.alves@ccc.ufcg.edu.br`

³ `matheus@ccc.ufcg.edu.br`

⁴ `thais.toscano@ccc.ufcg.edu.br`

Resumo The abstract should briefly summarize the contents of the paper in 15–250 words.

Keywords: First keyword · Second keyword · Another keyword.

1 Introdução

2 Verificação

3 Validação

4 Priorização de Requisitos

4.1 Definição

A priorização é uma forma de lidar com demandas conflitantes por recursos limitados. Estabelecer a prioridade relativa de cada capacidade de produto permite planejar a construção para fornecer o maior valor com o menor custo, com isso o mesmo ajuda o projeto a fornecer o máximo valor comercial o mais rápido possível dentro das restrições do projeto.

Até mesmo um projeto de tamanho médio pode ter dezenas de requisitos do usuário e centenas de requisitos funcionais. Para mantê-lo gerenciável, escolha um nível apropriado de abstração para a priorização - recursos, casos de uso, histórias de usuários ou requisitos funcionais. Dentro de um caso de uso, alguns fluxos alternativos podem ter uma prioridade mais alta que outros.

Alan Davis (2005) indica que a priorização bem sucedida requer uma compreensão de seis questões: As necessidades dos clientes; A importância relativa dos requisitos para os clientes; O momento em que as capacidades precisam ser entregues; Requisitos que servem como predecessores para outros requisitos e outros relacionamentos entre requisitos; Quais requisitos devem ser implementados como um grupo; O custo para satisfazer cada requisito.

Há requisitos que devem ser implementados juntos ou em uma sequência específica, pois não fará sentido implementar algo que não possa ser desfeito e/ou atualizado, por exemplo, você pode acabar escrevendo código para, digamos, aceitar pagamentos com cartão de crédito sem verificar se o cartão é válido, rejeitando cartões que foram relatados como roubados ou manipulando outras exceções.

4.2 Técnicas

Priorizar requisitos não é algo simples e devido a necessidade cada vez maior do uso de metodologias ágeis no desenvolvimento de softwares surgiram algumas técnicas que auxiliam nesse processo.

In or Out É caracterizado pela formação de um grupo que tem interesse no projeto, o mesmo produz uma lista de requisitos e vão tomando decisões binárias: *Este requisito está dentro ou fora?*

Pairwise comparison and rank ordering O grupo envolvido no projeto atribui uma lista com uma sequência de prioridades exclusivas para cada requisito. Após concluída, faz-se comparações entre pares dentre todos eles, para que no final seja percebido qual membro de cada par tem maior prioridade.

Three-level scale Essa abordagem de priorização agrupa os requisitos em três categorias: alta, média e baixa. Há alguns fatores para que sejam estabelecidos qual nível os requisitos estão. Os requisitos de alta prioridade são importantes e urgentes. Alternativamente, obrigações contratuais ou de conformidade podem ditar que um requisito específico deve ser incluído, ou pode haver razões comerciais convincentes para implementá-lo prontamente. Requisitos de prioridade média são importantes mas não são urgentes. Os requisitos de baixa prioridade não são importantes, nem urgentes, e ainda há uma quarta categoria onde diz que os requisitos parecem urgentes para alguns interessados, mas eles não são realmente importantes para alcançar os objetivos de negócios.

Ao executar uma análise de priorização com a escala de três níveis, você precisa estar ciente das dependências de requisitos. Você terá problemas se um requisito de alta prioridade depender de outro classificado com prioridade mais baixa e, portanto, planejado para implementação posteriormente.

MoSCoW As quatro letras maiúsculas representam quatro possíveis classificações de prioridade para os requisitos em um conjunto (IIBA 2009):

- **Must:** O requisito deve ser satisfeito para que a solução seja considerada um sucesso;

- Should: O requisito é importante e deve ser incluído na solução, se possível, mas não é obrigatório para o sucesso;
- Could: É uma capacidade desejável, mas que pode ser adiada ou eliminada. Implemente-o somente se o tempo e os recursos permitirem;
- Won't: isso indica um requisito que não será implementado no momento, mas pode ser incluído em uma versão futura.

Esse esquema altera a escala de três níveis para uma escala de quatro, ele não oferece nenhuma justificativa para tomar a decisão sobre como classificar a prioridade de um determinado requisito em comparação a outros. Ele também possui ambiguidade quanto ao tempo, particularmente quando se trata da classificação "Won't". "Não" pode significar "não no próximo lançamento" ou "nunca", com isso, esta técnica não é recomendada.

\$100 (hundred-dollar) Esta técnica consiste em dar à equipe de priorização 100 dólares imaginários para trabalhar, os mesmos alocam esses dólares para "comprar" itens que eles gostariam de ter implementado do conjunto completo de requisitos do candidato. Eles ponderam os requisitos de prioridade mais alta alocando mais dólares a eles. Se um requisito for três vezes mais importante para um interessado do que outro requisito, ela atribui talvez nove dólares ao primeiro requisito e três dólares ao segundo, desta forma faz com que os participantes realizem suas próprias alocações de dólares e, em seguida, somem o número total de dólares atribuídos a cada requisito para ver quais deles, coletivamente, têm a prioridade mais alta.

4.3 Priorização baseada em valor, custo e risco

Uma maneira definitiva e rigorosa de relacionar o valor do cliente aos recursos do produto proposto é com uma técnica chamada Implantação da Função de Qualidade, ou QFD (Cohen 1995). Os participantes típicos no processo de priorização incluem o gerente de projetos ou analista de negócios, que lidera o processo, arbitra conflitos e ajusta os dados de priorização recebidos dos outros participantes, se necessário, e os representantes do cliente que fornecem as classificações de benefício e penalidade.

Representantes de desenvolvimento, que fornecem as classificações de custo e risco:

1. Listar em uma planilha todos os recursos, casos de uso, fluxos de casos de uso, histórias de usuários ou requisitos funcionais que você deseja priorizar uns contra os outros. Usamos recursos no exemplo. Todos os itens devem estar no mesmo nível de abstração: não misture requisitos funcionais com recursos, casos de uso ou histórias de usuários.
2. Classificar de 1 a 9 os benefícios relativos ao cliente de cada requisito (1 indica que ninguém consideraria útil; 9 significa que seria extremamente valioso).
3. Classificar de 1 a 9 as penalidades caso aquele requisito não seja incluído no projeto (1 significa que ninguém ficará chateado se estiver ausente; 9 indica uma desvantagem séria).

4. A planilha calcula o valor total de cada característica como a soma de suas pontuações de benefício e penalidade. A planilha soma os valores de todos os recursos e calcula a porcentagem do valor total que vem de cada um dos recursos (a coluna Valor%).
5. Classificar de 1 a 9 os custos relativos da implementação, pelos desenvolvedores, (1 – rápido e fácil – a 9 – demorado e caro). Os desenvolvedores estimam as classificações de custo com base na complexidade do recurso, na extensão do trabalho da interface do usuário necessária, na capacidade potencial de reutilizar o código existente, na quantidade de testes necessária e assim por diante.
6. Classificar de 1 a 9 o risco técnico relativo. O risco técnico é a probabilidade de não obter o recurso logo na primeira tentativa. Uma classificação de grau 1 significa que o risco é baixíssimo, enquanto que 9 indica sérias preocupações sobre a viabilidade, a falta de conhecimento necessário sobre a equipe, o uso de ferramentas e tecnologias desconhecidas, ou a preocupação com a quantidade de complexidade escondida dentro do requisito.
7. Depois de inserir todas as estimativas na planilha, deve-se calcular um valor de prioridade para cada recurso usando a seguinte fórmula:

$$priority = value\% / (cost\% + risk\%) \quad (1)$$

8. Por fim, deve-se classificar a lista de recursos em ordem decrescente por prioridade calculada, a coluna mais à direita. Os recursos no topo da lista têm o balanço de valor, custo e risco mais favorável e, portanto, todos os outros fatores sendo iguais devem ter a prioridade mais alta.

A utilidade desse modelo de prioridade é limitada pela capacidade da equipe de estimar o benefício, a penalidade, o custo e o risco de cada item

Referências

1. Wiegers, K., Beatty, J.: Software Requirements. 3^a ed. (2013).