

# Planejamento de Releases de Software

## Trabalho de Implementação

Prof. Altino Dantas

30 de abril de 2019

### Disciplina Inteligência Computacional

Este documento apresenta um resumo sobre o problema do Planejamento de Releases de Software apresentado por Basílio-Neto (2016) e deve ser utilizado como principal referência para o trabalho de implementação da referida disciplina.

## 1 Modelagem do Planejamento de *Release*

Existem diversas modelagens para o Planejamento de *Releases* e o que as distinguem, de modo geral, são os fatores considerados por cada uma delas. A seguir, serão explicitados os conjuntos, elementos e atributos utilizados para modelar os fatores considerados neste trabalho, bem como a representação de solução adotada e formulação matemática como problema de otimização.

### 1.1 Requisitos e *Releases*

O conjunto  $R = \{r_1, r_2, r_3, \dots, r_N\}$  é tido como o conjunto dos  $N$  requisitos que devem ser alocados em  $P$  *releases*, representadas por  $K = \{k_1, k_2, k_3, \dots, k_P\}$ . Considerando que todo requisito  $r_i$  possui um custo e um risco de implementação a ele associados,  $custo_i$  e  $risco_i$  são utilizados, respectivamente, para definir tais características. O custo de implementação é um valor (geralmente dado em horas de trabalho ou montante financeiro) que representa o consumo total dos recursos necessários para implementação do requisito.

O risco é definido em termos da análise de impacto do risco para o negócio do cliente *versus* a sua probabilidade de ocorrência. O Quadro 1 mostra a quantificação do risco utilizada neste trabalho.

Os requisitos podem ainda possuir determinados relacionamentos entre si, os quais restringem a aceitação de soluções que não satisfaçam tais condições. Nesta abordagem são consideradas as

Tabela 1: Quantificação do Risco através da Análise de Impacto *versus* Probabilidade de Ocorrência.

Impacto no Negócio	Probabilidade de Ocorrência		
	Baixo	Médio	Alto
Baixo	1	2	3
Médio	4	5	6
Alto	7	8	9

relações de *precedência* e de *acoplamento* entre as funcionalidades. No primeiro caso, a implementação de um requisito  $r_i$  está condicionada a implementação de outro requisito  $r_j$ . Já uma relação de acoplamento indica que dois requisitos distintos devem ser implementados numa mesma *release*. Assim, ambos os relacionamento podem ser representados pela matriz binária  $M_{N \times N}$  que representa todos os  $N$  requisitos com seus respectivos dependentes, da seguinte forma:

$$M = \begin{pmatrix} m_{1,1} & m_{1,2} & \cdots & v_{1,i} \\ m_{2,1} & m_{2,2} & \cdots & v_{2,i} \\ \vdots & \vdots & \ddots & \vdots \\ m_{j,1} & m_{j,2} & \cdots & v_{N,N} \end{pmatrix},$$

onde, se  $m_{ij} = 1$ , o requisito  $r_i$  depende do requisito  $r_j$  e quando  $m_{ij} = m_{ji} = 1$ , os requisitos  $r_i$  e  $r_j$  devem obrigatoriamente ser implementados numa mesma *release*.

Além disso, cada *release*  $k_q$  possui uma restrição de orçamento, aqui denotada por  $s_q$ . Em face tais definições, as soluções para o Planejamento de *Releases* devem **favorecer a alocação antecipada dos requisitos com risco elevado sem, no entanto**, desrespeitar as limitações do orçamento de cada *release*.

## 1.2 Clientes

Posto que muitos projetos de software possuem vários interessados no produto final, consideremos  $C = \{c_1, c_2, c_3, \dots, c_M\}$  como sendo o conjunto de tais envolvidos, onde  $M$  é o total destes. Dessa forma, a matriz  $V_{m \times n}$  é definida para armazenar o valor de importância que cada interessado  $c_j$  estabelece para cada requisito  $r_i$ , as colunas representam os  $M$  interessados e os  $N$  requisitos são representados pelas linhas, como segue:

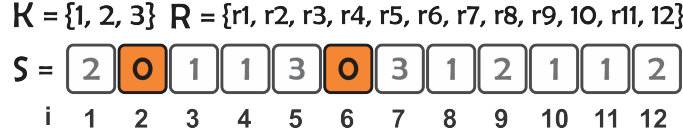
$$V = \begin{pmatrix} v_{1,1} & v_{1,2} & \cdots & v_{1,n} \\ v_{2,1} & v_{2,2} & \cdots & v_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ v_{m,1} & v_{m,2} & \cdots & v_{M,N} \end{pmatrix}.$$

Além disso, é utilizado um fator de peso  $w_j$ , associado a cada interessado  $c_j$ , para mensurar o nível de relevância que ele possui junto à organização responsável pelo desenvolvimento do software.

### 1.3 Representação da Solução

Tendo as definições dos conjunto dos requisitos e das *releases* acima expostas, a abordagem da presente pesquisa adota a representação de uma solução para o planejamento das *releases* como sendo um vetor  $S = \{x_1, x_2, x_3, \dots, x_N\}$ , onde  $x_i \in \{0, 1, 2, \dots, P\}$ . Se  $x_i = 0$ , significa que o requisito  $r_i$  não está alocado em *release* alguma, caso contrário, o requisito está alocado na *release*  $k_q$ , sendo  $q = x_i$ . A Figura 1 exemplifica um cenário no qual os requisitos  $r_2$  e  $r_6$  não estão alocadas, enquanto os requisitos  $r_1$ ,  $r_9$  e  $r_{12}$  estão selecionado para implementação na *release*  $K_2$ .

Figura 1: Representação de solução para Planejamento de *Releases* com vetor de inteiros



Esta é uma representação adequada para métodos de resolução baseados na manipulação da solução, como por exemplo, algoritmos Genéticos.

### 1.4 Formulação matemática do Planejamento de *Releases*

Uma condição para tratar um problema como problema de otimização é a existência de uma função capaz de avaliar soluções candidatas distinguindo-as qualitativamente. Assim, uma função capaz de avaliar soluções para o problema do planejamento de *releases*, considerando as definição anteriores poderia ser:

$$score(S) = \sum_{i=1}^N (importancia_i \times (P - x_i + 1) - risco_i \times x_i) \times y_i, \quad (1)$$

onde  $y_i \in \{0, 1\}$  é uma variável de decisão que contém 1 se o requisito  $r_i$  estiver alocado em algum *release*, ou seja,  $x_i \neq 0$ , e 0 caso contrário. A propriedade  $importancia_i$  contém a soma ponderada das importâncias que cada cliente  $c_j$  especificou para o requisito  $r_i$ , calculada da seguinte forma:

$$importancia_i = \sum_{j=1}^M w_j \times V(c_j, r_i). \quad (2)$$

Percebe-se que o valor função *score* é maior à medida que requisitos mais importantes e com menores riscos são alocados nas primeiras *releases*. Assim, consegue-se maximizar o valor de negócio da alocação e minimizar o risco global do projeto.

Portanto, a resolução do planejamento de *releases* consistem em:

$$\begin{aligned}
& \text{maximizar } score(S), \\
& \text{sujeito a: } 1) \sum_{i=1}^N custo_i \times f_{i,q} \leq s_q, \forall q \in \{1, \dots, P\}, \\
& \quad 2) x_i \geq x_j, \forall i, j \mid m_{ij} = 1,
\end{aligned}$$

onde  $f_{i,q}$  indica se o requisito  $r_i$  está alocado na *release*  $k_q$  e  $m_{ij}$  indica se  $r_i$  depende de  $r_j$ . Assim, 1) representa a restrição de orçamento, assegurando que a soma dos custos dos requisitos alocados para cada *release*  $k$  não ultrapasse o orçamento  $s_k$  disponível e 2) garante as restrições de precedência e acoplamento.

## 2 Instância do problema

Consideremos um cenário fictício no qual uma determinada empresa de desenvolvimento de software precisa desenvolver uma aplicação web para 3 clientes que atuam no comércio de *FastFood* e prestam serviço de entrega a domicílio.

Após o levantamento de requisitos e negociações com os clientes, ficou definido que o software seria liberado em três versões, tendo a organização desenvolvedora um orçamento de R\$ 125,00 para cada período correspondente às *releases*. Naturalmente, os clientes possuem opiniões divergentes em relação à urgência de implementação das funcionalidades e, por consequência, atribuíram diferenciados valores de importância para cada requisito.

Ademais, por diferentes razões, tais *stakeholders* possuem relevâncias distintas para a empresa de desenvolvimento. A Tabela 2 apresenta mais detalhes sobre esse contexto.

Tabela 2: Exemplo de informações que compõem uma instância do PR.

3 Releases. Orçamento: R\$ 125.00 cada		Relevância		3	4	2
#	Descrição	Custo	Risco	Cliente 1	Cliente 2	Cliente 3
0	Efetuar Pedido	R\$ 60,00	3	10	10	5
1	Buscar Produtos	R\$ 40,00	6	8	10	6
2	Sugestão de Estabelecimentos Próximos	R\$ 40,00	2	6	4	8
3	Definir várias Formas de Pagamento	R\$ 30,00	6	5	9	1
4	Compartilhar nas Redes Sociais	R\$ 20,00	4	7	7	5
5	Emitir <i>Feedback</i> sobre aplicativo	R\$ 20,00	8	8	6	2
6	Login através de Redes Sociais	R\$ 25,00	9	6	6	4
7	Executar em Múltiplas Plataformas	R\$ 70,00	7	9	8	3
8	Sistema de Recomendação	R\$ 50,00	6	6	7	5
9	Cadastro de Produtos	R\$ 20,00	6	10	10	7

Diante do cenário apresentado, o gerente de projetos da empresa precisa tomar a decisão sobre quais requisitos devem ser alocados em cada uma das três *releases* da aplicação. Todavia, considerar todas as informações levantadas e realizar uma análise manual seria uma tarefa exaustiva e fatigante. Assim, o referido profissional poderia optar por utilizar, como ferramenta para auxiliá-lo neste processo, a abordagem proposta neste trabalho.

Ao fazer uso da técnica de otimização considerando todas as definições apresentadas na seção 1, uma possível solução para esta instância pode ser vista na Figura 2.

Figura 2: Representação de possível solução para a instância apresentada.

Si	1	2	3	2	1	1	0	3	2	1
i	0	1	2	3	4	5	6	7	8	9

Um leitor atento já percebeu que as três últimas colunas da Tabela 2 podem ser resumida a uma única coluna pela aplicação da Equação 2. Isto geraria a vantagem de não ser necessário executar tal equação sempre que se desejar calcular o valor da função **score**.

---

## Referências

Basílio-Neto, A. D. 2016. Planejamento de Release Baseado em Otimização Interativa Através da Formalização das Preferências do Tomador de Decisão. Mestrado Universidade Estadual do Ceará, Fortaleza.