O Impacto de Unidades de Software Não Essenciais e Relações de Precedência Flexíveis sobre o Valor de Projetos de Software

Antonio Juarez Alencar[1], **Rafael Alcemar**[1], **Eber Assis Schmitz**[1], **Alexandre Luis Correa**[1], **Angélica F. S. Dias**[1]

[1]Instituto Tércio Pacitti de Aplicações e Pesquisas Computacionais
Universidade Federal do Rio de Janeiro

juarezalencar@dcc.ufrj.br, rafael.alcemar@gmail.com, eber@ce.ufrj.br
alexcorr@yahoo.com, angelica@nce.ufrj.br

Abstract. *This paper evaluates the impact of nonessential minimum marketable features modules (NMMF) and nonessential architectural elements (NAEs) on software projects, shows that the value-creation path of these self-contained software units may be quite different from that of essential software units, and discusses the impact of early NMMFs and NAEs identification on the value of software projects to business and the deployment of business strategies.*

Resumo. *Este artigo avalia o impacto das minimum marketable features modules (NMMF) e architectural elements não essenciais (NAEs) em projetos de software. Mostra que a criação de valor destas unidades de software pode ser bem diferente das unidades de software essenciais, e discute o impacto da identificação prévia de NMMFs e NAES sobre o valor de projetos de software para o negócio e para a implantação de estratégias de negócio.*

1. INTRODUÇÃO

Apesar do papel de destaque que a tecnologia da informação(TI) desempenha no cenário corporativo, o financiamento de projetos de desenvolvimento de *software* no ambiente altamente competitivo no qual as empresas fazem negócios se tornou uma questão central tanto para gerentes quanto para profissionais de tecnologia [Wu et al. 2007]. Como consequência, se torna cada vez mais difícil conseguir financiamento para projetos de desenvolvimento de *software* que não propiciem um valor claramente definido para o negócio [Denne and Cleland-Huang 2005].

Em sintonia com estas idéias, muitas propostas têm sido apresentadas para trazer disciplina financeira para o desenvolvimento de *software*. Enquanto algumas propostas são amplamente baseadas em métricas de avaliação financeira de projetos, tais como o valor presente líquido, retorno do investimento, taxa interna de retorno e, mais recentemente, teoria de opções reais, outras têm uma visão mais holística do desenvolvimento de *software* e defendem o uso de métricas baseadas em análise multivariada tais como *Strategy-to-Bottom-Line Value Chain*, *Multi-layer evaluation process* e *Information Economics* [Grembergen 2001].

Contudo, todas essas tentativas falham em reconhecer que a priorização e a modularizacão de requisitos têm um papel fundamental na construção do valor do *software*. Enquanto os requisitos satisfeitos por uma unidade de *software* são cruciais para a determinação do seu valor, a ordem na qual essas unidades são implementadas indicam

quão cedo esse valor pode ser apropriado. Uma exceção notável é apresentada por Dennne e Cleland-Huang [Denne and Cleland-Huang 2005] que sugerem o uso de uma análise financeira abrangente com a finalidade de maximizar o valor de projetos de *software* compostos por *minimum marketablede features modules* (MMFs), ou seja, unidades de *software* que contém pequenos conjuntos de funcionalidades que têm valor para o negócio. Em seu trabalho, Dennne e Cleland-Huang mostram que a ordem de implementação dos MMFs pode alterar substancialmente o valor destes projetos.

Contudo Denne e Cleland-Huang (*op. cit.*) falham em não reconhecer que (a) nem sempre todas as unidades de *software* são essenciais ao desenvolvimento de um projeto de *software*; (b) se uma unidade de *software* não for essencial a um projeto de *software*, o seu desenvolvimento pode ou não ser realizado pelo gerente de projetos durante o ciclo de vida do *software*; (c) o valor de unidades de *software* não-essenciais pode variar de acordo com o conjunto de unidades que precederam o seu desenvolvimento e (d) quando implementados, ao invés de criar o seu próprio fluxo de caixa, as unidades de *software* não-essenciais podem contribuir para o valor de um *software* influenciando de forma positiva o valor de unidades de *software* essenciais.

Este artigo é um passo à frente no preenchimento desta lacuna, revelando o valor de MMFs não essenciais (NMMFs) e de elementos arquiteturais não-essenciais (NAEs), cujo o processo de criação de valor pode ser diferente das MMFs tradicionais. Além disso, este artigo mostra como a combinação de relações de precedência flexíveis, NAEs e NMMFs pode ser usada para aumentar ainda mais o valor do *software*.

2. ARCABOUÇO CONCEITUAL

2.1. Casos de Uso

Caso de uso é uma linguagem de especificação gráfica e textual criada no final dos anos 1980's por Ivar Jacobson, que descreve como um conjunto de atores (o iniciador de interações) usa um sistema para alcançar um objetivo que tem valor para o negócio, e como o sistema ajuda estes atores a alcançar os seus objetivos [Bittner and Spence 2002]. A Figura 1 apresenta uma especificação de caso de uso de um sistema de controle de empréstimos. Na Figura 1 "Cliente" é um ator e as elipses são casos de uso, cujos significados são descritos na Tabela 1.

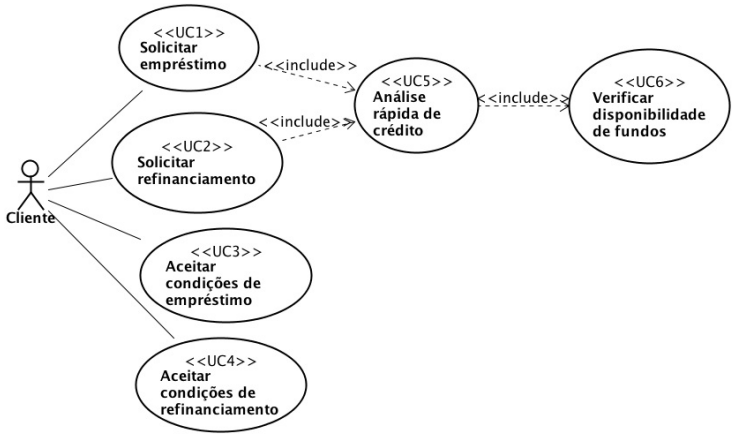

Figura 1. Um modelo de caso de uso de um sistema de controle de empréstimos.

Tabela 1. Descrição de caso de uso

Caso de Uso		
Id	**Nome**	**Descrição**
UC_1	Solicitar empréstimo	Coleta os dados necessários para conceder um empréstimo a um cliente
UC_2	Solicitar refinanciamento	Coleta os dados necessários para conceder o refinanciamento de um empréstimo existente
UC_3	Aceitar condições de empréstimo	Permite que o cliente aceite ou recuse as condições do empréstimo propostas pelo cedente
UC_4	Aceitar condições de refinanciamento	Permite que o cliente aceite ou recuse as condições de refinanciamento propostas pelo cedente
UC_5	Análise rápida de crédito	Verifica a probabilidade de um cliente pagar um empréstimo de acordo com os valores e as datas das parcelas
UC_6	Verificar disponibilidade de fundos	Verifica se o cedente possui os fundos necessários para conceder um empréstimo a um determinado cliente

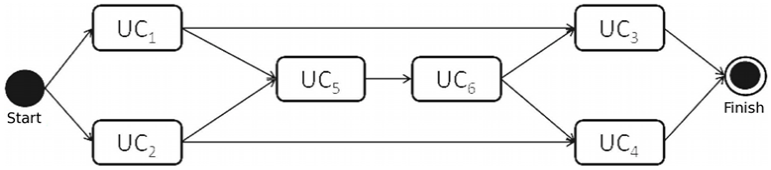

Figura 2. Diagrama de precedência do sistema de controle de empréstimo

Os casos de uso são independentes de tecnologia. Assim, casos de uso não incluem detalhes sobre *interfaces*, uso de bancos de dados e design de telas, podendo ser aplicados em várias fases do processo de engenharia de requisitos[Denney 2005].

2.2. Minimum Marketable Features

De acordo com Denne e Cleland-Huang [Denne and Cleland-Huang 2004] *minimum marketable features modules*, as MMFs, são módulos com pequenos conjuntos de funcionalidades que podem ser entregues de forma rápida e que criam valor para o negócio. Embora uma MMF seja uma unidade auto-contida, muitas vezes ela só pode ser desenvolvida depois que outras partes do projeto tiverem sido finalizadas. Estas partes do projeto podem ser outras MMFs ou a infra-estrutura arquitetural, ou seja, o conjunto de funcionalidades básicas que não oferecem valor direto aos clientes, mas que são necessárias às MMFs.

A própria infra-estrutura arquitetural pode ser decomposta em elementos autocontidos que podem ser entregues separadamente. Estes elementos, chamados de *architectural elements* (AEs), permitem que a arquitetura seja entregue de acordo com a demanda, reduzindo ainda mais o investimento inicial necessário para se executar um projeto. Entretanto, o valor total criado para o negócio por um software constituído de vários MMFs interdependentes, cada um com seu próprio fluxo de caixa e restrições de precedência, depende da ordem de desenvolvimento destes MMFs. Por exemplo, a Figura 2 apresenta o diagrama de precedência do conjunto de casos de uso da Figura 1. Neste exemplo em particular $UC_1, \ldots, UC_5$ são MMFs e UC_6 é um AE. As razões pelas quais estes casos de uso são considerados MMFs e AEs são objeto de discussão da Seção 3.

No diagrama apresentado na Figura 2, uma seta sai de UC_1 para UC_2, isto é, $UC_1 \rightarrow UC_2$, indica que o desenvolvimento do módulo UC_1 deve ser concluído antes que o de

UC_2 comece. O nó inicial (*Start*) e o final (*Finish*) servem apenas para marcar o começo e o término do projeto, respectivamente, e por isso, possuem duração e custo de desenvolvimento iguais a zero. A Tabela 2 mostra todas as possíveis seqüências de desenvolvimento para o sistema de controle de empréstimos, considerando que (a) cada unidade de *software* demora somente um período para ser desenvolvida; (b) somente uma unidade de *software* pode ser desenvolvida de cada vez; (c) a primeira unidade de *software* é desenvolvida no período 1; (d) não existe nenhum intervalo entre o fim do desenvolvimento de uma unidade de *software* e o começo do próximo.

Tabela 2. Opções de planejamento

Opções de Agendamento	Periodo					
	1	**2**	**3**	**4**	**5**	**6**
1	UC_1	UC_2	UC_5	UC_6	UC_3	UC_4
2	UC_1	UC_2	UC_5	UC_6	UC_4	UC_3
3	UC_2	UC_1	UC_5	UC_6	UC_3	UC_4
4	UC_2	UC_1	UC_5	UC_6	UC_4	UC_3

A Tabela 3 mostra o fluxo de caixa não descontado de cada MMF e AE apresentado na Figura 2. Por exemplo, de acordo com a Tabela 3, UC_1 requer um investimento inicial de US$ 50 mil. Uma vez que o seu desenvolvimento termina UC_1 produz uma série de retornos positivos até o décimo-quinto período, quando o sistema de empréstimo consignado torna-se obsoleto e é substituído por uma nova e mais avançada ferramenta.

Tabela 3. Use-case cash-flow elements

Cash Flow Elements (US$ 1,000)					
Id	**Period**				
	1	**2**	**3**	**4..14**	**15**
UC_1	-50	50	70	100	50
UC_2	-70	20	28	40	20
UC_3	-30	500	700	1,500	1,000
UC_4	-40	200	280	600	200
UC_5	-80	90	120	180	90
UC_6	-10	0	0	0	0

Uma vez que não é apropriado executar operações matemáticas sobre valores monetários em diferentes instantes de tempo sem considerar uma taxa de juros, é necessário calcular o seu fluxo de caixa descontado [Groppelli and Nikbakht 2006]. A Tabela 4 mostra o somatório do fluxo de caixa descontado de cada MMF da Figura 2, considerando uma taxa de juros de 2% por período. Este somatório é o valor presente líquido (VPL) de todos os elementos do fluxo de caixa de uma MMF. Para facilitar o entendimento, os valores apresentados na Tabela 4 foram arrendodados para o valor inteiro mais próximo. Os demais valores apresentados neste artigo seguem a mesma convenção.

Obviamente, nem todas as MMFs podem ser desenvolvidas no primeiro período. O grafo de precedência apresentado na Figura 2 indica que apenas o UC_1 e o UC_2 podem ser desenvolvido no primeiro período. Uma vez que neste exemplo cada MMF precisa de exatamente um período para ser desenvolvido, o UC_5 não pode ser desenvolvido até o terceiro período. Além disso, cada seqüência de MMFs produz o seu próprio VPL. Por exemplo, a seqüência $UC_1 \rightarrow UC_2 \rightarrow UC_5 \rightarrow UC_6 \rightarrow UC_3 \rightarrow UC_4$, produz um VPL de \$17.564 mil. Já a seqüência $UC_2 \rightarrow UC_1 \rightarrow UC_5 \rightarrow UC_6 \rightarrow UC_3 \rightarrow UC_4$, produz um VPL de \$17.601 mil.

Tabela 4. Use-case net-present value

Net Present Value (US$ 1,000)						
Id	**Period**					
	1	**2**	**3**	**4**	**5**	**6**
UC_1	1,045	987	894	802	712	623
UC_2	368	346	309	274	239	204
UC_3	16,001	14,944	13,537	12,157	10,804	9,478
UC_4	6,221	5,950	5,388	4,836	4,296	3,766
UC_5	1,885	1,781	1,613	1,447	1,285	1,126
UC_6	-10	-10	-10	-9	-9	-9

3. EXEMPLO DE APLICAÇÃO

As considerações apresentadas neste artigo são introduzidas com a ajuda de um exemplo inspirado no mundo real[1]. Empréstimos consignados são empréstimos de utilidade geral, de baixo risco e baixa taxa de juros concedidos a funcionários qualificados de empresas associadas a uma instituição financeira, nos quais os pagamentos são feitos em parcelas deduzidas de seus salários [Peterson 2008]. Neste sentido, considere uma instituição financeira que faz empréstimos consignados a seus clientes. Para o propósito deste artigo, esta organização será chamada de DINHEIRO RÁPIDO, or DR.

Assim que um pedido de um empréstimo é recebido, a DR calcula a probabilidade do cliente pagar o empréstimo solicitado de acordo com valores e datas pré-estabelecidas. Em seguida, a DR verifica se há fundos suficientes para conceder o empréstimo que o cliente está solicitando. Finalmente, são apresentadas ao cliente as condições nas quais um empréstimo consignado pode ser concedido, se houver. Neste ponto, o cliente pode optar por aceitar ou recusar a oferta do empréstimo.

Para oferecer uma resposta competitiva adequada aos movimentos recentes da concorrência no mercado de empréstimos consignado e desenvolver ainda mais seus negócios de empréstimos, a DR decidiu criar um novo sistema de empréstimo consignado para dispositivos móveis baseado na Internet. A empresa acredita que se agir rapidamente, além de aumentar o seu faturamento consideravelmente, esse sistema pode redefinir favoravelmente o cenário competitivo dos negócios de empréstimos consignado. A Figura 1 apresenta um modelo com os casos de uso do novo sistema de empréstimo consignado.

3.1. Determinando Como Cada Unidade de *Sofware* Pode Gerar Receita

A Tabela 1 descreve o significado e o tipo de cada caso de uso apresentado na Figura 1. Observa-se que todos os cinco primeiros casos de uso listados na tabela são unidades de *software* autônomas e que estas unidades geram receita para a DR da seguinte forma:

- *UC_1 e UC_2* - pergunta a cada cliente que solicita um empréstimo consignado se concorda em receber ofertas de novos produtos da DR e de seus parceiros comerciais. Novas campanhas de marketing podem ser aplicadas aos clientes que concordam, gerando novas oportunidades de vendas para a DR e receita na forma de taxas devidas pelo uso da base de clientes da DR por empresas parceiras;
- *UC_3 e UC_4* - assim que um cliente aceita uma oferta de empréstimo ou refinanciamento, é gerada receita na forma de juros e

[1]O exemplo foi inspirado em um projeto desenvolvido por uma grande instituição financeira sul-americana, que solicitou que seu nome não fosse divulgado.

- UC_5 - os resultados do caso de uso "Análise rápida de crédito" são usados para enriquecer a base de dados de clientes da DR. As informações de análise de crédito são particularmente importantes para campanhas de marketing que dependem da saúde financeira dos clientes, especialmente as campanhas que permitem que o cliente pague por um produto ou serviço em parcelas. As empresas associadas à DR pagam uma taxa especial para acessar a base de dados enriquecida.

Portanto os casos de uso UC_1, UC_2, $\cdots$, UC_5 são MMFs. Por outro lado, UC_6, "Verificar a disponibilidade de fundos" é o único AE entre os casos de uso na Figura 1, já que apresenta um serviço essencial ao comportamento desejado do sistema. No entanto, nem os clientes nem as empresas associadas estão dispostas a pagar por este serviço.

3.2. Planejando a Implementação do Sistema

Embora a equipe da DR responsável pelo projeto de construção do sistema de empréstimo consignado tenha considerado inicialmente adotar o diagrama de precedência da Figura 2, logo perceberam que o comportamento da unidade UC_5 é independente do tipo de empréstimo que está sendo solicitado, isto é: novo empréstimo ou refinanciamento. Portanto, o desenvolvimento de UC_5 pode começar quando o desenvolvimento de UC_1 ou UC_2 estiver completo, ou mesmo quando o desenvolvimento de ambas estiver completo.

Considerando que esta nova visão das dependências requeridas por UC_5 é uma opção real, e que as opções poderem mudar substancialmente o valor de projetos de *software* [Fichman et al. 2005], a equipe de projeto da DR decidiu alterar o diagrama de precedência apresentado na Figura 2. A Figura 3 mostra o diagrama de precedência atualizado para o projeto de construção do sistema de empréstimo consignado.

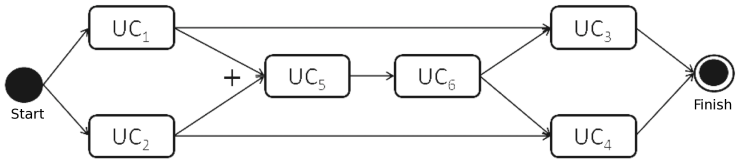

Figura 3. Diagrama de precedência do projeto de sistema de empréstimo consignado.

Observa-se que o requisito de precedência flexível do UC_5 é apropriadamente sinalizado pela presença do símbolo "+" (ou inclusivo) no diagrama. Também é importante observar que, de acordo com as condições atuais de mercado, determinadas através de pesquisa de mercado encomendada pela equipe de analistas de negócios da DR, as unidades apresentadas na Figura 3 formam o conjunto de unidades de *software* que são essenciais para o desenvolvimento do sistema de empréstimo consignado, ou seja este é o menor conjunto de unidades autônomas que tem alguma chance de apresentar retorno a longo prazo ao investimento a ser feito pela DR em seu desenvolvimento.

A Tabela 5 indica todas as sequências possíveis de desenvolvimento do sistema de empréstimo consignado. Existem ao todo quatorze possibilidades. É importante mencionar que como as unidades UC_1, UC_2, $\cdots$, UC_6 são todas essenciais ao comportamento desejado do sistema, todas devem ser desenvolvidas.

3.3. Avaliação de Diferentes Opções de Planejamento

A Tabela 6 mostra os elementos de fluxo de caixa não-descontados de cada MMF e AE no modelo apresentado na Figura 3 conforme estimado pela equipe de projeto da DR. Uma

vez que a unidade UC_5 gera receita de acordo com o número de clientes existentes na base de dados de clientes da DR (vide Seção 3.2), seus elementos de fluxo de caixa variam de acordo com o contexto no qual UC_5 é desenvolvido. Quanto maior o número de clientes na base de dados, maior a receita gerada por UC_5. Portanto

- Se UC_1 preceder o desenvolvimento de UC_5, o que é indicado por $UC_1 \rightarrow UC_5$ na Tabela 6, então UC_5 produz US$ 60 mil no segundo período, US$ 90 mil no terceiro período, US$ 130 no quarto período, e assim sucessivamente;
- Se UC_2 preceder o desenvolvimento de UC_5, ou seja, $UC_2 \rightarrow UC_5$, então o UC_5 produz valores menores em cada período. Produz US$ 20 mil no segundo período, US$ 30 mil no terceiro, US$ 50 mil no quarto, e assim sucessivamente;
- Contudo, se tanto UC_1 quanto UC_2 precederem o desenvolvimento de UC_5, ou seja, (UC_1,UC_2) $\rightarrow UC_5$, então o valor produzido por UC_5 em cada período atinge os seus valores mais altos. UC_5 produz US$ 90 mil no segundo período, US$ 120 mil no terceiro período, US$ 180 no quarto período, e assim sucessivamente.

Em todas as circunstâncias, o investimento necessário pelo UC_5 permanece o mesmo, isto é US$ 80 mil.

A Tabela 7 relaciona as opções de planejamento apresentadas na Tabela 5 para os seus respectivos VPLs. Por exemplo, na sétima opção, UC_1 é desenvolvido primeiramente. Então, UC_5, UC_6, UC_3, UC_2 e UC_4 são desenvolvidas no segundo, terceiro, quarto, quinto e sexto períodos, respectivamente, produzindo um VPL de US$18.460 mil, que é o VPL mais alto entre as opções de planejamento e, como resultado, a escolha lógica da equipe de projetos da DR.

Tabela 5. Opções de Planejamento

Opções de Planejamento	**Período**					
	1	**2**	**3**	**4**	**5**	**6**
1	UC_1	UC_2	UC_5	UC_6	UC_3	UC_4
⋮	⋮	⋮	⋮	⋮	⋮	⋮
7	UC_1	UC_5	UC_6	UC_3	UC_2	UC_4
⋮	⋮	⋮	⋮	⋮	⋮	⋮
14	UC_2	UC_5	UC_6	UC_4	UC_1	UC_3

Tabela 6. Elementos de fluxo de caixa não-descontados das unidades de *sofware*

Elementos de Fluxo de Caixa (US$ 1.000)					
Unid. de *Software*	**Período**				
	1	**2**	**3**	**4 to 14**	**15**
UC_1	-50	50	70	100	50
UC_2	-70	20	28	40	20
UC_3	-30	500	700	1,500	1,000
UC_4	-40	200	280	600	200
$UC_1 \rightarrow UC_5$	-80	60	90	130	60
$UC_2 \rightarrow UC_5$	-80	20	30	50	20
(UC_1,UC_2) $\rightarrow UC_5$	-80	90	120	180	90
UC_6	-10	0	0	0	0

3.4. Lidando com Unidades de *Software* Não-Essenciais

Antes de fazer uma opção de planejamento com base em um modelo que considere apenas unidades de *software* essenciais, a equipe de projetos da DR decidiu examinar o efeito das unidades não essenciais, que podem melhorar ainda mais a qualidade dos serviços prestados pela DR. A Tabela 8 descreve as unidades de *software* não essenciais. Há somente uma MMF não essencial (NMMF) entre as unidades de *software* não essenciais, ou seja, UC_7, "Análise detalhada de crédito", que enriquece ainda mais a base de dados de clientes da DR com informações que abrem oportunidades para novas campanhas de marketing, tanto da DR, quanto de empresas associadas. Como a receita gerada pela unidade UC_7 depende do número de clientes aprovados, é razoável afirmar que quanto mais clientes tiverem seu crédito analisado, maior tende a ser a receita gerada.

Assim sendo, a receita gerada por UC_7 é influenciada diretamente pelo desenvolvimento das unidades UC_1,"Solicitar empréstimo", e UC_2, "Solicitar refinanciamento", assim como UC_5, "Análise rápida de crédito". A Tabela 9 apresenta os elementos de fluxo de caixa de UC_7. A receita gerada por esta unidade de *software* em particular se dá através das empresas associadas à DR que pagam uma taxa especial para acessar a base de dados de clientes enriquecida.

Surpreendentemente, esta não é a única forma em que UC_7 contribui para o valor do projeto de construção do sistema de empréstimo consignado. Se implementada, a unidade "Análise detalhada de crédito" aumenta o número de pedidos de empréstimos e refinanciamentos aprovados. Quanto mais pedidos de empréstimos e refinanciamentos forem aprovados, maior o retorno produzido pelas unidades "Aceitar condições de empréstimo" e "Aceitar condições de refinanciamento". Portanto, a unidade de *software*

Tabela 7. VPL da Opção de Planejamento

Opção de Planejamento	VPL (US$ 1,000)
1	17,564
2	16,769
⋮	⋮
7	18,460
⋮	⋮
14	15,814

Tabela 8. Descrições das unidades de *software* não essenciais

Unidade de *software*		
Id	**Nome**	**Descrição**
UC_7	Análise detalhada de crédito	Executa uma análise de crédito mais detalhada em pedidos que foram rejeitados pela "Análise rápida de crédito", a fim de conceder o empréstimo a estes clientes, caso isso seja possível
UC_8	Contratar seguro de vida	Para conceder os empréstimos, inclui o custo de um seguro de vida em pedidos de empréstimo rejeitados pela "Análise rápida de crédito" devido a baixa expectativa de vida desses clientes
UC_9	Lidar com empréstimos aprovados	Determina a data mais provável na qual a DR pode conceder um pedido de empréstimo ou refinanciamento que foi rejeitado por "Verificar disponibilidade de fundos" devido a uma falta de fundos temporária

"Análise detalhada de crédito" também contribui para o aumento da receita gerada por estas duas outras unidades de *software*.

Tabela 9. Elementos de fluxo de caixa de UC_7

Elementos de Fluxo de Caixa (US$ 1.000)					
Unidade de *Software* Id	**Período**				
	1	**2**	**3**	**4 to 9**	**10**
$UC_1 \rightarrow UC_7$	-90	65	75	90	65
$UC_2 \rightarrow UC_7$	-90	25	30	35	25
$(UC_1,UC_2) \rightarrow UC_7$	-90	90	100	130	90

Por outro lado, "Lidar com empréstimos aprovados" e "Contratar seguro de vida" são unidades autônomas cujos serviços nem os clientes, nem as empresas associadas desejam pagar. Portanto, são de fato elementos de arquitetura não essenciais (NAEs). Apesar destas duas unidades não gerarem receita por si mesmas, contribuem para o valor do sistema de empréstimo consignado influenciando a receita gerada por outras unidades.

De modo semelhante a "Análise detalhada de crédito", "Lidar com empréstimos aprovados" e "Contratar seguro de vida" aumentam o número de pedidos de empréstimos e refinanciamentos aprovados. Como resultado, aumentam a receita gerada por "Aceitar condições de empréstimo" e "Aceitar condições de refinanciamento". A Tabela 10 apresenta o impacto estimado do desenvolvimento de unidades de *software* não essenciais sobre a receita gerada por unidades de *software* essenciais.

Tabela 10. O impacto estimado de unidades de *software* não essenciais sobre a receita gerada por unidades de *software* essenciais

		Unidade de *Software* Essencial	
		UC_3 (Aceitar condições de empréstimo	**UC_4** (Aceitar condições de refinanciamento)
Unidade de Software Não- Essencial	**UC_7** (Efetuar análise detalhada de crédito)	+20%	+15%
	UC_8 (Contratar seguro de vida)	+15%	+10%
	UC_9 (Lidar com empréstimos aprovados)	+25%	+20%

3.5. Replanejando a Implemantação do Software

A Figura 4 apresenta o novo diagrama de precedência do sistema de empréstimo consignado. Este diagrama considera a existência de unidades de *software* essenciais e não essenciais. A linha tracejada ao redor da identificação da unidade tal como UC_7 ou UC_8 indica uma unidade de *software* não essencial, que pode ser desenvolvida ou não.

Contudo, se um caso de uso não essencial for desenvolvido, as dependências que este cria devem ser satisfeitas. Por exemplo, se nem UC_7 nem UC_8 forem desenvolvidas, o desenvolvimento de UC_6 pode começar imediatamente após o desenvolvimento do UC_5 estar completo. Porém, se UC_7 for desenvolvida e UC_8 não for, o desenvolvimento de UC_6 só pode começar quando o desenvolvimento de UC_5 e UC_7 estiver completo, e assim por diante. A Tabela 11 apresenta as opções de planejamento do sistema de

empréstimo consignado, levando em consideração a existência de casos de uso essenciais e não essenciais. Há ao todo 144 diferentes opções de planejamento.

Tabela 11. Novas Opções de Planejamento

#	Período								
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
1	UC_1	UC_2	UC_5	UC_6	UC_3	UC_4			
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
81	UC_2	UC_1	UC_5	UC_7	UC_8	UC_6	UC_3	UC_4	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
144	UC_2	UC_5	UC_7	UC_9	UC_6	UC_1	UC_8	UC_4	UC_3

3.6. Qual Opção de Planejamento Apresenta o VPL Mais Alto?

A Tabela 12 exibe os elementos de fluxo de caixa não-descontados de cada MMF, AE, NMMF e NAE apresentados na Figura 4, conforme estimado pela equipe de projetos da DR.

Tabela 12. Novos elementos de fluxo de caixa da unidade de *software*

Novos Elementos de Fluxo de Caixa (US$ 1.000)					
Unid. de *Software*	**Período**				
Id	**1**	**2**	**3**	**4 to 9**	**10**
UC_1	-50	50	70	100	50
UC_2	-70	20	28	40	20
UC_3	-30	500	700	1,500	1,000
$UC_7 \rightarrow UC_3$	-30	600	840	1,800	1,200
$UC_8 \rightarrow UC_3$	-30	625	875	1,875	1,250
$UC_9 \rightarrow UC_3$	-30	575	805	1,725	1,150
$(UC_7,UC_8) \rightarrow UC_3$	-30	725	1,015	2,175	1,450
⋮	⋮	⋮	⋮	⋮	⋮
UC_9	-10	0	0	0	0

A Tabela 13 exibe o VPL de cada MMF, AE, NMMF e NAE apresentado na Figura 4 de acordo com o período no qual é desenvolvido, levando em consideração uma taxa de juros de 2% por período. Observa-se que o número de períodos que se leva

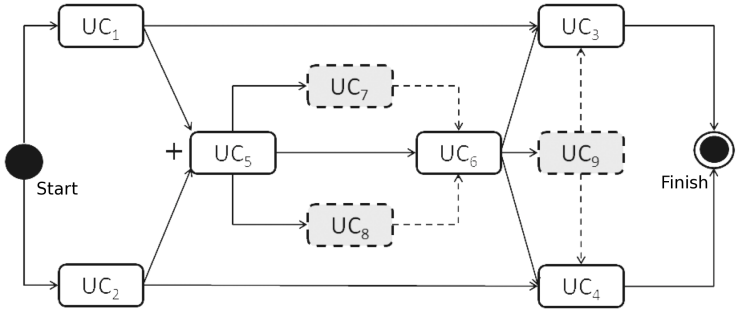

Figura 4. Um diagrama de precedência contendo unidades de *software* essenciais e não essenciais.

Tabela 13. VPLs de unidades de *software* essenciais e não essenciais

Valor Presente Líquido (US$ 1.000)					
Unid. de *Software*	**Período**				
Id	**1**	**2**	**3**	$\cdots$	**10**
UC_1	1,045	987	894	$\cdots$	369
UC_2	368	346	309	$\cdots$	105
UC_3	16,001	14,944	13,537	$\cdots$	5,653
$UC_7 \rightarrow UC_3$	19,207	17,939	16,250	$\cdots$	6,788
$C_8 \rightarrow UC_3$	20,009	18,688	16,928	$\cdots$	7,072
$UC_9 \rightarrow UC_3$	18,406	17,190	15,572	$\cdots$	6,504
$(UC_7,UC_8) \rightarrow UC_3$	23,215	21,683	19,641	$\cdots$	8,208
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
UC_9	-10	-10	-10	$\vdots$	-9

para desenvolver todo o sistema aumentou de seis para nove períodos, devido à possível introdução de MMFs e AEs não essenciais (vide Tabela 11).

Entre todas as opções de planejamento apresentadas na Tabela 11, a opção mais lógica para a DR é desenvolver as unidades de *software* que compõem o sistema de empréstimo consignado na seguinte ordem: $UC_2 \rightarrow UC_1 \rightarrow UC_5 \rightarrow UC_7 \rightarrow UC_8 \rightarrow UC_6 \rightarrow UC_3 \rightarrow UC_4$, que produz um VPL de US$ 19.651. Observe que esta opção de planejamento não contempla o desenvolvimento do UC_9, "Lidar com empréstimo aprovado". De acordo com as estimativas geradas pela equipe de projetos da DR, o custo de desenvolvimento do UC_9 não compensa o seu impacto positivo sobre a receita gerada pelos UC_3 e UC_4.

4. CONCLUSÃO

A Tabela 14 compara os valores das diferentes alternativas consideradas pela equipe de projetos da DR para o desenvolvimento do sistema de controle de empréstimo consignado. Observe que, geralmente, o que as unidades de *software* não essenciais e as relações de precedência flexíveis fazem é permitir que gerentes de projeto adiem o desenvolvimento de unidades de *software* menos lucrativas até que se tornem totalmente necessárias. Como resultado, o valor de trilhas de desenvolvimento mais rentávies são obtidas e podem ser apropriadas antecipadamente.

Tabela 14. Alternativas consideradas pela DR para o desenvolvimento do sistema de controle de empréstimo consignado

#	Referência		Flexibilidade	VPL da Escolha Lógica (US$ 1.000)
	Mod. do Caso de Uso	Diag. de Preced.		
1	Figura 1	Figura 2	Nenhuma	US$17.564
2	Figura 1	Figura 3	Relação de precedência de dependência entre os UC_1, UC_2 e UC_5 é flexibilizada	US$18.460
3	—	Figura 4	Relação de precedência entre os UC_1, UC_2 e UC_5 é flexibilizada. NMMFs e NAEs são apresentados	US$19.651

Como resultado, é seguro afirmar que a introdução de relações de precedência flexíveis e a presença de NMMFs e NAEs entre as unidades de software essenciais pode

aumentar o valor de negócio de projetos de software. Uma vez que o objetivo esperado para o projeto de software está claramente definido, as unidades de software não essenciais são aquelas que podem ser retiradas do projeto sem prejudicar o objetivo traçado anteriormente[Pohl et al. 2010, Valente 2009].

Apesar da considerável quantidade de pesquisa disponível na literatura sobre o impacto da flexibilidade na avaliação, seleção e no gerenciamento de projetos de *software* [Benaroch et al. 2006, Wu and Ong 2008, Fichman et al. 2005], surpreendentemente, a introdução de flexibilidade através de unidades de *software* não essenciais e relações de precedência flexíveis é pouco explorada. O objetivo deste artigo é preencher essa lacuna. Tendo consciência do valor potencial de NMMFs e relações de dependência flexíveis, as empresas podem seguir as idéias de Denne e Cleland-Huang [Denne and Cleland-Huang 2005] e desenvolver projetos maiores e mais complexos a partir de um investimento relativamente menor.

Referências

Benaroch, M., Shah, S., and Jeffery, M. (2006). On the valuation of multistage information technology investments embedding nested real options. *Journal of Management Information Systems*, 23(1):239–261.

Bittner, K. and Spence, I. (2002). *Use Case Modeling*. Addison-Wesley.

Denne, M. and Cleland-Huang, J. (2004). The incremental funding method: Data-driven software development. *Software, IEEE*, 21(3):39– 47.

Denne, M. and Cleland-Huang, J. (2005). Financially informed requirements prioritization. In *Proceedings of the 27th International Conference on Software Engineering (ICSE)*, pages 710–711, St. Louis, Missouri, USA. ACM Press.

Denney, R. (2005). *Succeeding with Use Cases*. Addison-Wesley.

Fichman, R. G., Keil, M., and Tiwana, A. (2005). Beyond valuation: Real options thinking in IT project management. *California Management Review*, 47(2):74–100.

Grembergen, W. V. (2001). *Information Technology Evaluation Methods and Management*. Idea Group Publishing.

Groppelli, A. A. and Nikbakht, E. (2006). *Finance*. Barron's, 5^{th} edition.

Peterson, C. L. (2008). Usury law, payday loans, and statutory sleight of hand. *Minnesota Law Review*, 92(4).

Pohl, K., Bckle, G., and van der Linden, F. J. (2010). *Software Product Line Engineering: Foundations, Principles and Techniques*. Springer, NY, USA.

Valente, P. (2009). *Goals Software Construction Process*. VDM.

Wu, L.-C. and Ong, C.-S. (2008). Management of information technology investment: A framework based on a real options and mean variance theory perspective. *Technovation*, 28(3):122–134.

Wu, S. D., Shi, C., and Gurbaxani, V. (2007). Investigating the risk-return relationship of information technology investment: Firm-level empirical analysis. *Management science*, 53(12):1829–1842.