# *COGNITIO*: UM PROCESSO PARA REUSO DE REQUISITOS

**Ceres Germanna Braga Morais**
Universidade do Estado do Rio Grande do Norte (UERN) e Instituto Federal do RN.
Mossoró – RN – Brasil. E-mail:ceres.morais@ifrn.edu.br
**Jezmael de Oliveira Basilio**
Projeto Ação Digital (PAD), Russas – CE – Brasil. E-mail:
jezmael@projetoacaodigital.com.br
**Lyrene Fernandes da Silva**
Universidade Federal do Rio Grande do Norte (UFRN). Natal – RN – Brasil
E-mail: lyrene@dimap.ufrn.br
**Thalles Robson de Barbalho**
Universidade do Estado do Rio Grande do Norte (UERN)
E-mail: thall3sr@gmail.com

## RESUMO

Este artigo apresenta um processo de reuso de requisitos, denominado Cognitio, o qual busca fornecer ganhos na produtividade e qualidade dos softwares gerados, de forma que os stakeholders se sintam motivados tanto para desenvolver software com reuso quanto para o reuso. Para a formação do repositório, organizamos as informações relacionadas aos requisitos através da linguagem Q7, e apresentamos através do framework NFR como o nosso processo pode ser aplicado.

Palavras Chaves – Cognitio, linguagem Q7.

# *COGNITIO*: A PROCESS FOR REUSE OF REQUERIMENTS

## ABSTRACT

This paper presents a process for reuse of requirements, called Cognitio, that seeking to provide improvement in productivity and quality of the generated software, so that stakeholders are motivated both to develop software for reuse and reuse them. To build the repository, we have organized the related information to the requirements using the language Q7, and we have presented it through the NFR framework the way our process will be applied.

Key-Words – Cignitio, language Q7.

## INTRODUÇÃO

Tendo em vista a necessidade de se desenvolver sistemas de qualidade e que atendam às especificações dos *stakeholders*, entra em foco a Engenharia de Requisitos (ER), cujo papel fundamenta-se em descobrir o que o software a ser gerado deverá proporcionar, através da análise e documentação dos seus requisitos [Nuseibeh e Easterbrook 2000].
No entanto, nem sempre a atividade de ER é valorizada durante o desenvolvimento de software, o que causa, na maioria das vezes, a elevação dos custos durante e após a sua implementação. Tentando evitar tal problema, entra em foco o reuso de requisitos o qual consiste em usar de forma sistemática documentos de requisitos existentes com o intuito de diminuir o esforço e aumentar a qualidade dentro do ciclo de vida do software [Villegas e Laguna 2001]. O reuso permite, assim, que a aplicação seja construída a partir de requisitos que já foram previamente especificados, validados e testados, o que resulta no ganho da produtividade e na qualidade do software criado.

Porém, para haver o reuso efetivo de requisitos, é extremamente importante que os engenheiros de requisitos e os desenvolvedores de software tenham acesso aos requisitos que podem ser reusados. Neste caso, torna-se indispensável a existência de processos e métodos que conduzam os engenheiros de requisitos a se preocuparem em definir seus requisitos de forma a permitir o seu posterior reuso.

Para o desenvolvimento do processo *Cognitio*, utilizamos a linguagem de organização de qualidades denominada Q7 [Leite et al 2005]. Essa linguagem oferece parâmetros de classificação, os quais facilitam a organização do repositório de requisitos, uma vez que os dados são representados de forma clara e precisa, viabilizando a busca e recuperação dos requisitos por parte dos interessados.

Para apresentamos como o processo é realizado, escolhemos a abordagem *Framework* NFR (Non-Functional Requirements) [Chung et al 2000] para modelar os requisitos, dado que expressa tanto requisitos funcionais quanto não funcionais, além de vir sendo utilizado para catalogar requisitos, estabelecendo o relacionamento e conflitos entre ele. O restante deste artigo está organizado da seguinte maneira: a Seção 2 mostra os trabalhos relacionados; a Seção 3 aborda o processo de reuso *Cognitio*; e a Seção 4 apresenta as nossas conclusões.

## TRABALHOS RELACIONADOS

Para melhor entendimento do *Cognitio* apresentamos a seguir alguns conceitos que são peculiares ao mesmo.

## REQUISITOS

A Engenharia de Requisitos (ER) é uma área que se preocupa com a elicitação de requisitos, sua documentação e respectivo gerenciamento [Ramos 1999], auxiliando a

modelagem e análise de sistemas, de forma que suas características sejam bem entendidas antes de serem implementadas [Robinson; Pawlowski e Volkov, 2003].

Requisitos são entendidos como condição ou capacidade de que um usuário necessita para solucionar um problema específico, ou para atingir um objetivo. Nesse artigo, utilizamos a definição de requisitos funcionais (RF) e não funcionais (RNF), sendo que os RFs expressam funções ou serviços que um software deve ou pode ser capaz de executar ou fornecer, enquanto que os RNFs são reconhecidos como os atributos de qualidade de um software.

## REUSO DE REQUISITOS

O reuso de requisitos consiste em uma abordagem para sistematizar o uso de documentos de requisitos existentes, com o objetivo de reduzir o esforço desprendido nas fases do ciclo de vida do software [Villegas e Laguna 2001], para o desenvolvimento de uma nova aplicação. Essa abordagem é útil para dar assistência e orientação ao engenheiro de requisitos, durante todo o processo de elicitação de requisitos [Rolland e Prakash 1999]. A redução do esforço através do reuso de requisitos se dá de duas formas: melhorando o processo de ER [Cybulski 1998] e dando suporte ao processo de desenvolvimento de software através do reuso de informações [Bellizona; Fugini e Percini 1992].

## ENGENHARIA DE REQUISITOS ORIENTADA A METAS

A Engenharia de requisitos orientada a metas está preocupada com o uso de metas para realizar as atividades do processo de engenharia de requisitos (elicitação, documentação, negociação, entre outras). Dentre as abordagens de modelagem de requisitos orientada a metas, temos o *framework* NFR, o qual provê os relacionamentos entre metas e softmetas. Para tanto, ele possui artefatos como softmetas, catálogos e grafos SIG (Softgoal Interdependency Graph).

Para a representação dos nossos modelos de requisitos utilizamos os grafos SIG. O grafo SIG possui os elementos meta, softmeta e *claim*; e os relacionamentos do tipo decomposição (que podem ser do tipo *e* e *ou*), operacionalização (que pode ser do tipo *make*, *help*, *hurt* e *break*) e argumentação (que consiste na ligação entre uma *claim* e um relacionamento de decomposição ou operacionalização). A Figura 1 apresenta um exemplo de um grafo SIG e seus elementos, definidos na cor vermelha.

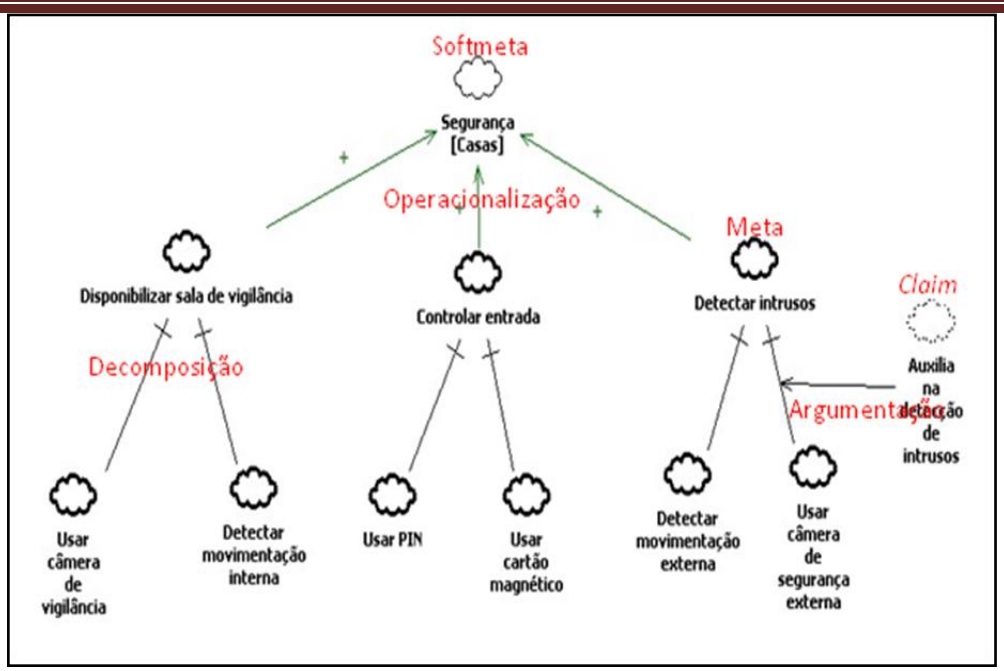


**Figura 1. Representação de um Grafo SIG**

As softmetas representam os RNFs; as metas geralmente representam os RFs; e as *claims* comportam-se como argumentos necessários para a ocorrência de um dado relacionamento. Logo, em nosso processo, utilizamos os grafos SIGs de forma compreendendo RFs como metas e RNFs como softmetas. De acordo com a Figura 1, a decomposição ocorre quando um elemento é refinado em outro do mesmo tipo, por exemplo, quando uma softmeta é refinada em outras softmetas; a operacionalização refina as softmetas em metas; a argumentação funciona como justificativa ou alerta do porque de uma determinada interdependência existir.

## LINGUAGEM Q7

Para realizar a classificação dos grafos, escolhemos a linguagem Q7 [Leite et al 2005], a qual consiste de sete questões representadas como 5W2H (*why*, *who*, *what*, *where*, *when*, *how* e *how much*). Esta linguagem foi desenvolvida a fim de assegurar tanto características de qualidade, quanto o relacionamento dessas características com descrições funcionais, dentro da abordagem de orientação a metas e de orientação a aspectos.

A linguagem Q7 é de suma importância para o nosso processo, haja vista que é capaz de representar conceitos tais como funções, pré-condições, contribuições, entre outros aspectos que contribuem para a classificação de requisitos a serem reusados.

Para o nosso escopo, a adaptação da linguagem Q7 ficou da seguinte forma (de acordo com o exemplo da Figura 1): *Why*: indica o requisito pai de um relacionamento (por exemplo, a softmeta "Segurança" é pai da meta "Detectar intrusos"); *Who*: indica o papel exercido por quem solicitou um dado requisito (por exemplo, "engenheiro de requisitos"); *What:* indica o contexto no qual cada requisito do grafo está inserido (por exemplo, o termo "[Casa]", da softmeta "Segurança"); *Where*: indica o domínio do grafo (por exemplo, o contexto "Segurança de casas"); *When:* indica uma espécie de justificativa da existência de um

determinado relacionamento, a qual chamamos de *claim* (representado por "auxilia na detecção de intrusos"); *How*: indica os filhos de um determinado relacionamento ("Detectar intrusos" é filho de "Segurança"); *How much*: indica o tipo de relacionamento existente entre um *why* e um *how* (o relacionamento entre "Detectar intrusos" e "Segurança" é uma "Operacionalização" do tipo *make*).

## PROCESSO DE REUSO DE REQUISITOS

Por meio de um estudo acerca dos assuntos apresentados na Seção anterior, foi gerado o processo *Cognitio.* Para compor o *Cognitio*, foram definidos alguns sub-processos os quais têm o intuito de atingir um objetivo comum que é fornecer o reuso de requisitos, buscando a qualidade dos requisitos e o incentivo à prática do reuso. Para um melhor entendimento, a Figura 2 apresenta a abordagem na notação BPMN (Business Process Modeling Notation) [BPMN 2009].

O processo é dividido em quatro *pools* distintos, denominados Modelagem de requisitos, Mecanismo de armazenamento e classificação, Mecanismo de busca e recuperação, e Mecanismo de versionamento, de forma que cada pool é responsável pelo desenvolvimento de seus respectivos sub-processos.

Para melhor explicar o processo de reuso de requisitos, tem-se a premissa que ele está sendo realizado pela primeira vez. Assim, o sub-processo inicial é Modelar grafo, realizado pelos *stakeholders*, no *pool* Modelagem de requisitos. Durante a modelagem, devem ser realizadas também as atividades de verificação e validação dos requisitos.

Após a modelagem, o mecanismo Armazenar grafos pode ser iniciado, uma vez que o usuário desejar guardar os requisitos modelados, gerando um repositório de dados.

Quando um grafo é armazenado, ocorre o sub-processo Classificar grafos com a linguagem Q7. Dessa forma, esse mecanismo reconhece as informações importantes do modelo de requisitos, e os classificam de acordo com os parâmetros da linguagem Q7. Paralelamente ao sub-processo de classificação, ocorre o sub-processo Controlar versão dos grafos. Após ocorrer o controle de versão, os grafos versionados e classificados são disponibilizados para a busca.

Neste momento, o sub-processo Buscar grafos de acordo com os parâmetros de classificação pode ser iniciado, a partir do Mecanismo de busca e recuperação. Ao buscar um grafo, o usuário tem as opções de recuperar ou não os grafos resultantes da busca. Caso ele não deseje recuperar o grafo, o sub-processo de busca é finalizado, caso contrário, o sub-processo Recuperar grafos é iniciado.

Ao recuperar um grafo, o usuário pode editá-lo ou não. Caso ele não deseje editar o grafo, o processo de recuperação é finalizado. Caso contrário, ele realiza as edições que achar necessárias. Verifica-se que a edição dos grafos não é definida no Cognitio, cabendo a realização da mesma aos *stakeholders*. Com a edição dos grafos, o usuário realiza uma nova modelagem dos mesmos, reiniciando, portando, o processo de reuso de requisitos.

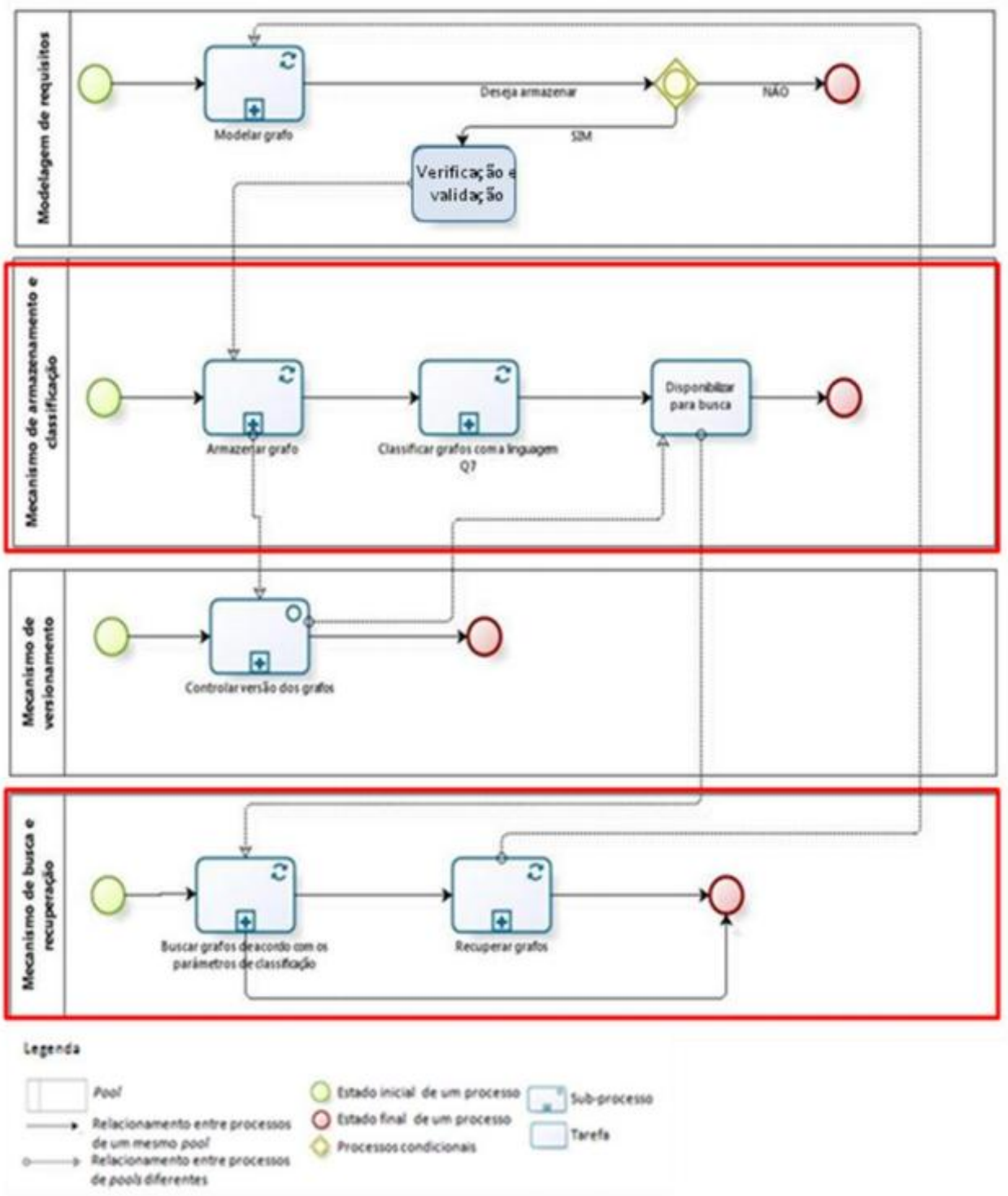


**Figura 2. Processo *Cognitio***

## CONCLUSÕES

A principal motivação para o desenvolvimento deste trabalho foi o fato da ER ser por muitas vezes negligenciada durante o processo de desenvolvimento de software. Com isto, ocorrem grandes custos aos desenvolvedores, tendo em vista as falhas na especificação do sistema. Dessa forma, com a aplicação do processo de reuso definido, é disponibilizado aos

engenheiros de requisitos e demais *stakeholders* uma fonte de informações para elicitação de requisitos, uma vez que o usuário pode utilizar o *Cognitio* para busca e recuperação de requisitos.

O trabalho desenvolvido tem como contribuição a definição de um processo de reuso de requisitos, apresentando-se, portanto, como fonte de conhecimento, que pode auxiliar durante a elicitação de requisitos, além da adaptação da linguagem Q7 não só para o escopo de reuso de RNFs, mas também para os RFs.

Entre as limitações, temos que até o momento, o *Cognitio* contempla apenas o armazenamento de requisitos modelados nos grafos SIG do framework NFR. Além disso, o Cognitio não consiste em verificar se os dados armazenados são de qualidade, isto é, se eles passaram pelas fases de elicitação, validação e verificação do processo de Engenharia de Requisitos. Logo, esta tarefa fica incumbida aos *stakeholders*, o que não se pode garantir que as mesmas são cumpridas, embora o *Cognitio* exija que as mesmas devam ser realizadas antes da ocorrência do armazenamento dos grafos no repositório.

Como trabalhos futuros, sugerimos a ampliação do processo para que o mesmo reconheça outros modelos de representação de requisitos, além do framework NFR; a verificação de outras formas de recuperação das informações contidas no repositório, isto é, que diagramas são indispensáveis, ou que outros dados são pertinentes para o auxílio do reuso de requisitos.

## REFERÊNCIAS


1. Bellizona, R.; Fugini, M. G.; Percini, B. (1992) **An environment for specification reuse**. Technical Report POLIMIUDUNIV. 92.E.2.9E.8.4, Ithaca.
2. BPMN. **Business Process Modeling Notation**. Disponível em: http://www.bpmn.org/. Acesso em: dezembro de 2009.
3. Chung, L.; Nixon, B.; Yu, E.; Mylopoulos, J. (2000) **Non-Functional Requirements in Software Engineering**. Kluwer Academic Publishers. Boston/Dordrecht/London.
4. Cybulski, J. L. (1998) **Patterns in software requirements reuse**. Technical report, Department of Information Systems. University of Melbourne.
5. Leite, J. C. S. P.; Yu, Y.; Liu, Y., Yu E. S. K.; Mylopoulos, J. (2005) **Quality-Based Software Reuse**. CAiSE-05, LNCS 3520, pp.535-550, Springer-Verlag.
6. Nuseibeh, B.; Easterbrook, S. (2000) **Requirements engineering: a roadmap**. In ICSE '00: Proceedings of the Conference on The Future of Software Engineering, New York, NY, USA. ACM Press. pág. 35.46.
7. Ramos, R. A. (1999) **Treinamento prático em UML**. Universo dos Livros Editora LTDA.
8. Robinson, W. N.; Pawlowski, S. D.; Volkov, V. (2003) **Requirements Interaction Management**. ACM Computing Surveys, Vol. 35, No. 2. p. 132-190.
9. Rolland, C.; Prakash, N. (1999) **From conceptual modelling to requirements engineering**. Technical Report Series 99-11, CREWS.

10. Villegas, O. L.; Laguna, M. A. (2001) **Requirements Reuse for Software Development**. RE 01 Doctoral Workshop. in 5th - IEEE International Symposium on Requirements Engineering. Toronto, Canada, 27–31.