Engenharia de Papéis e o Processo XP: uma Proposta de Integração através do Jogo do Planejamento

Ludmila A. Pedrosa[1,2], Gustavo H. M. B. Motta[1]

[1]Programa de Pós Graduação em Informática (PPGI) – UFPB
CEP 58035-000 – João Pessoa – PB – Brasil

[2]Simplestec Informática Ltda. – João Pessoa – PB – Brasil

ludmila.pedrosa@simplestec.com.br, gustavo@di.ufpb.br

Abstract. *This paper presents the proposal of a methodology for integrating the requirements of role-based access control (RBAC) obtained through role engineering to the eXtreme Programming (XP) software process. This integration is made through an extension of the XP planning game to include the steps of the role engineering that will result in new outputs called Role-related User stories, the tests cards related to Role-related User stories and the concrete RBAC model.*

Resumo. *Este trabalho propõe uma metodologia que integra os requisitos do controle de acesso baseado em papéis (RBAC), obtidos através da engenharia de papéis, ao processo de desenvolvimento eXtreme Programming (XP). A integração é realizada através de uma extensão do jogo do planejamento XP para incluir os passos da engenharia de papéis que resultarão em novas saídas chamadas de Role-related User stories, os cartões de testes relacionados às Role-related User stories e o modelo RBAC concreto.*

1. Introdução

A segurança de um sistema de informação é um fator de grande importância para muitas organizações. Mesmo os sistemas mais simples possuem sua complexidade inerente e softwares disponíveis, íntegros e de fácil utilização são cada vez uma exigência mais forte. A necessidade da segurança da informação se deve a diversos fatores, incluindo a rápida proliferação da internet e dos sistemas via web, a grande quantidade de plataformas heterogêneas que passaram a ser conectadas via sistemas distribuídos e ao alto volume de informações sensíveis que trafegam nas redes das organizações e que precisam estar protegidas (Joshi et al., 2001), tudo isso associado a um crescimento potencial de usuários diretos e indiretos dos sistemas computacionais. Dessa forma, a segurança da informação visa atender tal demanda, buscando garantir que os sistemas sejam capazes de prevenir que atacantes alcancem seus objetivos, ou seja, o acesso ou uso não autorizado aos computadores e suas redes (Howard, 1997).

Muitos modelos têm sido apresentados com o propósito de controlar o acesso dos usuários às aplicações. Os modelos clássicos de controle de acesso são o discricionário e compulsório, porém estes carecem dos requisitos necessários para a definição e administração de políticas de acesso que demandem um grande número de usuários e recursos, particularmente nas organizações coorporativas (Joshi et al., 2001). Nos últimos anos, o modelo que vem melhor se adequando aos sistemas de informação das grandes organizações é o controle de acesso baseado em papéis – RBAC (Ferraiolo et al., 2001, ANSI/INCITS 359, 2004). O RBAC regula o acesso dos usuários aos

objetos protegidos com base nos papéis que eles exercem numa organização. Os papéis denotam funções que descrevem a autoridade e a responsabilidade concedidas aos usuários. As autorizações para acessar objetos não são associadas diretamente a usuários, mas aos papéis, de acordo com as atribuições pertinentes. Com o controle de acesso baseado em papéis, uma organização pode definir a sua política de acesso estabelecendo as permissões de acordo com as funções que os usuários exercem na organização, tornando-se assim, uma alternativa atrativa, por ser uma solução simples e flexível, características essenciais para as empresas atuais. Ademais, o RBAC é politicamente neutro, podendo suportar os modelos compulsório ou discricionário, dentre outros.

Atualmente existem no mercado várias ferramentas que implementam diversas extensões do modelo RBAC. Porém, o uso de ferramentas, apesar de importante, não é o suficiente. Na verdade, é necessária a utilização de uma metodologia que auxilie na definição e execução de uma política de segurança para controle de acesso. Nesse sentido, uma área que vem recebendo bastante atenção nos últimos anos é a engenharia de papéis (Coyne, 1995; Epstein; Shandu, 2001; Roeckle et al., 2000; Neumann; Strembeck, 2002; Shin et al., 2003). Podemos entender a engenharia de papéis como sendo uma metodologia para definição de papéis e atribuição de respectivas permissões e restrições, permitindo assim, identificar e explicitar os objetos usados no controle de acesso, dentro da existência implícita de papéis numa organização (Coyne, 1995). Antes que um modelo RBAC concreto possa ser implementado tecnicamente, as atividades da engenharia de papéis precisam acontecer, ou seja, o modelo RBAC concreto é o resultado final do processo da engenharia de papéis.

Infelizmente, muitas abordagens da engenharia de papéis apresentadas são definidas meramente em bases *ad hoc* ou tratam apenas de parte do processo (Neumann; Strembeck, 2002). Por outro lado, não consideram os requisitos de controle de acesso de forma integrada ao processo de software. A necessidade dessa integração fica explícita quando observamos a necessidade eminente do levantamento dos requisitos de controle de acesso durante a fase inicial do projeto, reduzindo o esforço de uma manutenção futura para que estes requisitos sejam atendidos. Portanto, pensar na definição de políticas de controle de acesso como parte integrante do processo de software, nos ajudará a identificar de forma antecipada e natural, a seqüência de eventos e necessidades que são importantes para a implantação e funcionamento adequado deste software, de acordo com a política de segurança estabelecida pela organização. A expectativa é que as boas práticas da engenharia de software, associadas às boas práticas da engenharia de papéis tragam contribuições para todo o processo de software.

Para tornar esta integração mais leve, nos preocupamos em identificar um processo de software que fosse mais adequado à flexibilidade exigida pelas organizações contemporâneas. Para tal, usaremos uma nova abordagem para o desenvolvimento de projetos conhecida como “métodos ágeis” (Fowler, 2005). Estes métodos baseiam-se na iteratividade, concentram-se nos requisitos e enfatizam a comunicação direta mais do que uma documentação pesada da abordagem tradicional para desenvolvimento de software. Dentre os vários processos ágeis, nós selecionamos o XP – *eXtreme Programming* (*EXtreme Programming*, 2007) pela forte influência humana do processo, característica comum a engenharia de papéis.

O objetivo deste trabalho é propor a integração da engenharia de papéis ao processo de software *eXtreme Programming*. Para tanto, nós estendemos uma das atividades processo XP, denominada Jogo do Planejamento, que identifica e prioriza os

requisitos de negócios, para incluir passos adicionais que resultam em novas saídas: as *Role-related User stories* e os cartões de testes relacionados às *Role-related User stories*. O primeiro passo corresponde à definição das estórias relacionadas aos papéis desempenhados por cada usuário do sistema e o segundo vai produzir os testes de aceitação das estórias relacionadas aos papéis que irão gerar o modelo RBAC a ser construído pelo processo. Nossa abordagem é derivada da proposta do Processo de Engenharia de Papéis Orientado a Cenários de Neumann e Strembeck (2002), da qual aplicamos e modificamos alguns processos, para depois integrá-los ao Jogo do Planejamento XP.

O restante do trabalho está organizado da seguinte forma. A seção 2 faz uma breve revisão da engenharia de papéis, seus modelos e principais abordagens analisadas. Na seção 3 apresentamos a proposta integração da engenharia de papéis ao processo de software XP e, por fim, a seção 4 discute as contribuições de nossa proposta frente aos trabalhos relacionados e apresenta as conclusões e os trabalhos futuros.

2. A Engenharia de Papéis e os Trabalhos Relacionados

A engenharia de papéis é o processo de definir papéis, permissões, restrições e a hierarquia de papéis em modelos de segurança RBAC. Os principais benefícios de uma engenharia de papéis bem realizada são: a possibilidade de implementação de políticas de segurança RBAC com um controle de acesso efetivo; simplificação na administração da política e a identificação dos papéis de usuários que auxiliarão a engenharia de requisitos a levantar as funcionalidades do sistema e as interfaces com os usuários.

Os trabalhos existentes em engenharia de papéis têm como principal objetivo a implementação e administração de papéis. Estes podem ser classificados em três modelos: *top-down*, *bottom-up* e híbrido (Shin et al., 2003). No modelo *top-down*, fazemos primeiramente a identificação dos papéis para, a partir deles, derivarmos as permissões. O modelo *bottom-up* parte da derivação de permissões para depois agrupá-las dentro de papéis. Já o modelo híbrido pode ser descrito como uma combinação dos modelos *top-down* e *bottom-up*.

Dentre os trabalhos usando o modelo *top-down*, podemos citar inicialmente Coyne (1995), que utiliza as atividades dos usuários em sistemas, com um alto nível de abstração, para identificar papéis. Roeckle (2000) usa o conceito de *role-finding* com o propósito de deduzir papéis das necessidades de negócios ou funções da organização e o conceito de "meta-modelo" RBAC para descrever a noção de papéis, suas relações com usuários e direitos de acesso. Shin e Cho (2003) apresentam um modelo *top-down* de engenharia de papéis baseado na informação, fazendo uma definição e análise do conhecimento de alto-nível de um sistema.

Thomsen et al. (1998) propuseram um framework RBAC para ambientes de rede, nos quais permissões são derivadas de objetos e seus métodos, e papéis são derivados de permissões, sendo classificado como um modelo *bottom-up*. Seu trabalho apresenta sete camadas abstratas para o gerenciamento da segurança, as quatro primeiras camadas pertencem à aplicação do desenvolvedor, que pode usar seu conhecimento para de criar componentes de segurança genérico. As três últimas camadas estão sob controle da administração do sistema, responsável pela customização da política de segurança da organização.

Epstein e Shandu (2000) propuseram um framework conceitual onde papéis podem ser definidos tanto da maneira *top-down,* quanto *bottom-up*. Eles estenderam o

modelo de referência RBAC96 (Shandu et al., 1996) introduzindo três camadas adicionais entre papéis e permissões: *jobs*, *work-patterns* e *tasks*. Kern et al. (2005) também apresentam um modelo híbrido para o ciclo de vida dos papéis, sendo um processo iterativo-incremental, onde quatro estágios são identificados: análise de papéis, projeto de papéis, gerência de papéis e manutenção de papéis.

Neumann e Strembeck (2002) apresentam um processo *bottom-up* da engenharia de papéis orientado a cenários, sendo esse, um dos trabalhos mais detalhados na área. O modelo usa cenários como fonte de derivação de permissões e suas características mais importantes são: prevê a mudança de escopo e os incorpora nos modelos de forma simples, tornando-o mais flexível; permite a detecção de permissões com diferentes granularidades; também permite o uso de uma ferramenta de software para gerar a hierarquia de papéis preliminar de maneira semi-automática e provê principalmente uma abordagem sistemática da engenharia de papéis.

3. Integrando a Engenharia de Papéis ao *eXtreme Programming*

A prática do jogo do planejamento XP (Astels et al., 2002) é utilizada na fase de definição do que deverá ser produzido (escopo/requisitos e prioridades), de quando será entregue (estimativa dos *releases*) e de quais serão os próximos passos (iterações). O jogo do planejamento envolve dois passos chave: planejamento de uma versão (*release*) e o planejamento de uma iteração (*iteration*).

Inicialmente é feito o planejamento de uma versão (*release planning*), onde o cliente define as necessidades e os programadores estimam a dificuldade em atender os pedidos. No XP, os requisitos são especificados nas chamadas estórias (*user stories*), as quais são escritas em cartões indexados. As estórias são expressas em frases pequenas que precisam ser mensuradas e testadas. Uma parte essencial do jogo do planejamento é a negociação, onde o time de desenvolvimento estima cada estória (*user stories*) em termos de um tempo de programação ideal. O cliente então decide qual estória de usuário é prioritária e deve ser completada. Baseado nas estimativas provenientes dos desenvolvedores, das prioridades dos clientes, do tempo do projeto e recursos disponíveis, desenvolvedores e clientes finalmente discutem e negociam o que deve ser implementado para criar uma liberação (*release*).

Algumas semanas após o esforço de desenvolvimento inicial, todos se reúnem para um planejamento de iteração (*iteration planning*), onde é possível descrever detalhadamente as tarefas, o cliente pode ver o que já foi feito e, conjuntamente com a equipe de desenvolvimento, definir as próximas atividades (inclusive mudando o planejamento da versão). É importante lembrar que não há sobressaltos se o cliente descobrir que algo estava faltando, ele escreve uma nova narrativa de uso, definem-se os custos, e o cliente decide se aumenta o escopo ou reduz outras funcionalidades. Por isso, é importante sempre implementar narrativas de uso, que são os elementos da iteração. A qualquer momento, existe um produto a ser apresentado. Os processos de planejamento de iteração e de versão continuam até a entrega final do produto.

O processo de planejamento do jogo XP pode ser visto como suficiente para a especificação dos requisitos de controle de acesso, se tais requisitos fossem considerados como outros quaisquer. Porém, anteriormente mostramos que a engenharia de papéis é uma área importante, onde este assunto é tratado de forma aprofundada. Neste sentido, nossa proposta integra as atividades da engenharia de papéis ao jogo do

planejamento XP, para o levantamento dos requisitos de controle de acesso baseado em papéis e a conseqüente construção de um modelo RBAC concreto.

Nossa proposta se baseia em alguns sub-processos estabelecidos por Newman e Strembeck (2002), que apresentam um processo completo para a engenharia de papéis. Entretanto, não utilizaremos cenários no sub-processo inicial da derivação do modelo RBAC, conforme foi proposto por Newman e Strembeck. Ao invés disso, usaremos os conceitos das estórias de usuários (*user stories*), que possuem um papel fundamental no jogo do planejamento do *eXtreme Programming*. Resumidamente, as estórias de usuários (*user stories*) são breves descrições informais dos requisitos funcionais do sistema, escritas pelos próprios clientes. Elas ilustram como o sistema pode ser usado para criar valores e provêem detalhe suficiente para facilitar o levantamento dos requisitos e estimativas de tempo na sua implementação. Através dos conceitos de estórias de usuários (*user stories*), proporemos novas saídas ao jogo do planejamento, através das *Role-related User stories* e os cartões de testes relacionados às *Role-related User stories*. Bem como, criaremos novos sub-processos para incorporá-los, juntamente com alguns sub-processos já propostos por Newman e Strembeck, ao jogo do planejamento do *eXtreme Programming*.

3.1. Jogo do Planejamento proposto para o Controle de Acesso Baseado em Papéis

O Jogo do Planejamento proposto para o controle de acesso baseado em papéis é composto por oito atividades maiores que possuem sub-processos próprios. Como descrito na Figura 1, as atividades de 1 a 5 formam um ciclo que é repetido até que o modelo das *Role-related User stories* e os Testes de Aceitação esteja completo, estas atividades fazem parte do planejamento de iteração (*iteration planning*) do Jogo do Planejamento. Não obstante, o processo completo (atividades de 1 a 8) é executado de maneira iterativa e incremental, onde cada iteração resulta num novo estágio evolucionário de diferentes modelos (ver seção 3.3). Cada ciclo deste (atividades de 1 a 8) faz parte do planejamento de uma versão (*release planning*) do jogo do planejamento.

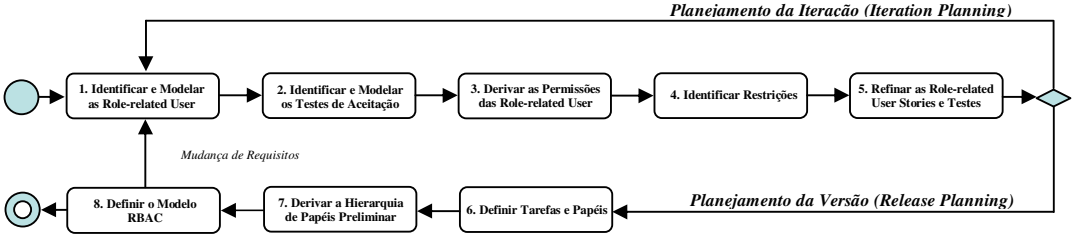

Figura 1: Visão Geral da extensão do Planejamento do Jogo para o Controle de Acesso Baseado em Papéis

As oito atividades do Jogo do Planejamento proposto para o RBAC, bem como, cada sub-processo próprio, são descritas nas subseções a seguir.

3.1.1. Identificar e Modelar as *Role-related User stories*

Essa atividade é executada em paralelo à identificação dos requisitos funcionais do sistema (*user stories*), através da construção das *Role-related User stories*. Dentro de cada *Role-related User Story* são identificados os sujeitos responsáveis por cada ação e são contadas estórias focadas nos acessos associados à função que aquele sujeito desempenha dentro da organização. As *Role-related User stories* são estórias dos

requisitos de controle de acesso para permitir a construção do modelo RBAC concreto.

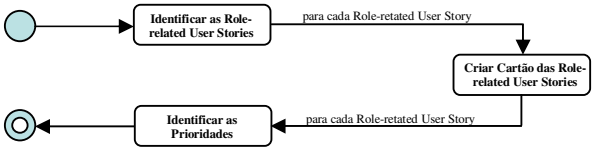

Figura 2: Sub-Processo de Modelagem das Role-related User Stories

O primeiro passo do sub-processo 1 (Figura 2) é identificar as *Role-related User stories* sensíveis do sistema, estas estórias são então documentadas em cartões indexados, cada um referente a uma estória específica. No terceiro passo da modelagem das *Role-related User stories*, elas são discutidas com a equipe de projetistas para garantir a sua relevância e com a equipe do cliente para a definição das prioridades. Observem que as prioridades atribuídas as *Role-related User stories* devem estar alinhadas com as prioridades das *user stories*. As *Role-related User stories* servem seqüencialmente para a derivação de cartões de aceitação, permissões, restrições e para a definição de papéis.

3.1.2. Identificar e Modelar os Testes de Aceitação

As *Role-related User stories* permitem compreender aquilo que deve ser criado em relação ao controle de acesso. Os testes de aceitação fornecem a confirmação de que criamos o que foi pedido. Devemos escrever o teste de aceitação com antecedência para que ele esteja pronto para fornecer essa confirmação quando for executado. Assim sendo, para cada *Role-related User Story*, deve haver pelo menos um teste de aceitação que mostre que o sistema demonstrou o comportamento esperado.

As atividades de identificar e modelar os testes de aceitação estão descritas no sub-processo da Figura 3. O procedimento consiste em visitar todas as *Role-related User stories* e primeiramente identificar as pré-condições ou cenários de teste e, depois, identificar as operações que são realizadas durante o teste. O terceiro passo trata de escrever as pós-condições ou aquilo que é verdadeiro após o teste ser concluído. Esse passo é chamado de seção de verificação. Por fim, devemos criar os cartões de testes de aceitação, que na nossa proposta, estão indexados e possuem uma referência cruzada para cada *Role-related User Story* relacionada.

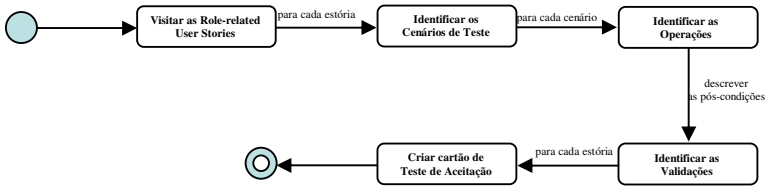

Figura 3: Sub-Processo de Modelagem dos Testes de Aceitação

3.1.3. Derivar as Permissões das *Role-related User stories*

O sub-processo de derivação de permissões é descrito na Figura 4. Durante a derivação das permissões, as *Role-related User stories* identificadas como prioritárias pela equipe de clientes são visitadas em ordem crescente do seu cartão de índice, juntamente com seus cartões de aceitação. Para identificar as permissões, podemos usar diretamente as ações dos sujeitos já identificadas nos cartões de aceitação e depois armazená-las em pares <operação, sujeito>. A meta das atividades executadas neste sub-processo é a

identificação das permissões para depois armazená-las no catálogo de permissões (ver Figura 10), que contém todas as permissões que foram detectadas nas *Role-related User stories*.

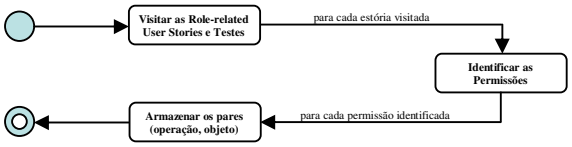

Figura 4: Sub-Processo de Derivação de Permissões

3.1.4. Identificar Restrições

O primeiro passo a ser realizado neste sub-processo é definir quais os tipos de restrições devem ser modeladas. Devemos seguir as permissões e, consequentemente, as restrições identificadas nos testes de aceitação associados às *Role-related User stories*.

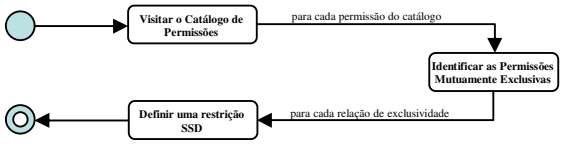

Figura 5: Sub-Processo de Identificação de Restrições

Definindo-se os tipos de restrições relevantes, a identificação pode começar. De fato, um sub-processo próprio para cada tipo de restrição (e. g., a separação de responsabilidades no modelo RBAC) seria necessário, mas como a seqüência de passos é a mesma, a Figura 5 apresenta um exemplo de identificação de permissões com restrições de separação de responsabilidades estática (restrições SSD) (Ferraiolo et al, 2001). As restrições são identificadas através da participação de pessoas experientes no domínio. Para a nossa proposta isto não será um problema, pois foi a própria equipe do cliente que participou da construção das estórias e dos testes de aceitação.

3.1.5. Refinar as *Role-related User stories* e Testes de Aceitação

Aqui as *Role-related User stories* e os Testes de Aceitação que foram construídos nas atividades 1 e 2 serão revisados e depois refinados. Podemos distinguir duas atividades essenciais neste sub-processo (Figura 6):

- Derivação: cada estória relacionada ao controle de acesso é revisada e, se for muito complexa, é dividida em duas ou mais *Role-related User stories*. A conseqüência desse passo é que também precisamos dividir os Testes de Aceitação da *Role-related User Story* original para que exista um para cada nova *Role-related User Story* criada. Na verdade estaremos aqui fazendo uma nova iteração das atividades dos sub-processos de 1 a 5.
- Generalização: as *Role-related User stories* são revisadas com o objetivo de identificar estórias similares. Para cada grupo de estórias similares, uma estória mais genérica é criada para substituí-las. Esta tarefa deve ser executada cuidadosamente, pois a prática *eXtreme Programming* exige que as estórias sejam as mais simples possíveis, mas nunca abstratas.

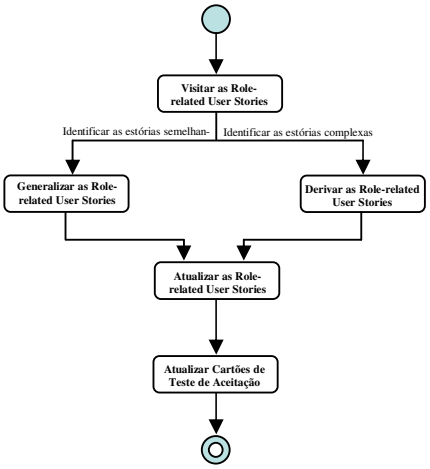

Figura 6: Sub-Processo Refinamento das Role-related User Stories e Testes de Aceitação

3.1.6. Definir Tarefas e Papéis

O objetivo principal deste sub-processo é seguir um enfoque *bottom-up* para derivar papéis de trabalho. Isso será feito diante do modelo de permissões (sub-processo 3) baseado nas *Role-related User stories* e nos Testes de Aceitação (sub-processos 1 e 2).

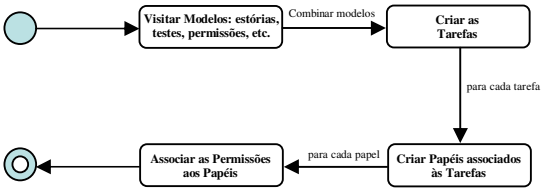

Figura 7: Sub-Processo de Definição de Tarefas e Papéis

Ao iniciarmos esta atividade, teremos que revisitar os modelos até agora gerados pelas atividades anteriores e identificar quais elementos podem ser combinados para executar uma tarefa completa. Após a identificação das tarefas, podemos então associa-las para definir papéis. As tarefas podem compor um ou mais papéis. Um papel é uma função de trabalho dentro do contexto de uma organização com algumas semânticas associadas, considerando a autoridade e a responsabilidade conferida ao usuário associado a ele. Como os papéis são formados por tarefas, que por sua vez, foram geradas dos modelos de permissões, restrições e estórias de controle de acesso, podemos identificar diretamente as permissões que precisam ser atribuídas a um papel. Concluindo assim, o sub-processo que pode ser visto na Figura 7.

3.1.7. Derivar a Hierarquia de Papéis Preliminar

Agora que já temos os papéis definidos e suas permissões associadas, podemos identificar os papéis que possuam semelhanças, o que significa procurar inicialmente por papéis que possuam exatamente as mesmas permissões que outros papéis (Figura 8). Estes papéis são agrupados, marcados para uma futura revisão e uma possível exclusão. No decorrer do processo, os papéis podem também ser equipados com permissões adicionais ou permissões podem ser removidas.

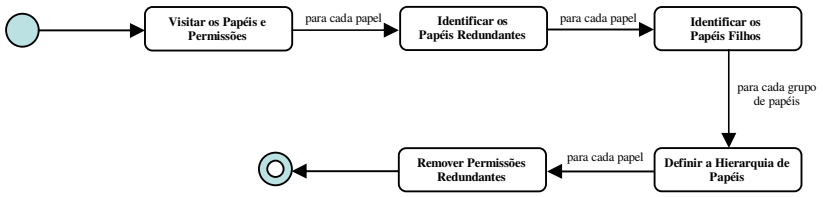

Figura 8: Sub-Processo de Derivação da Hierarquia de Papéis Preliminar

Portanto, antes que a hierarquia de papéis seja definida, uma hierarquia de papéis preliminar deve ser construída, observando-se as permissões que cada papel possui. No final da derivação da hierarquia de papéis, as permissões redundantes devem ser removidas. Isso significa que devemos remover todas as permissões que estão associadas ao papel e quase sempre isso será aplicado ao papel filho. Após a finalização deste passo, a hierarquia de papéis preliminar já se encontra construída.

3.1.8. Definir o Modelo RBAC

Neste sub-processo, a hierarquia de papéis preliminar, o catálogo de permissões e o catálogo de restrições são usados como entrada para a definição concreta do modelo RBAC. Os papéis ainda marcados como redundantes no sub-processo anterior são removidos, novos papéis e restrições são definidos, e as hierarquias de papéis são mescladas ou separadas. Estes passos são repetidos até o modelo RBAC estar completo, isto é, até que os engenheiros que são os responsáveis por esta atividade, juntamente com a equipe de clientes, decidirem que o modelo RBAC é adequado. A Figura 9 descreve a ordem das atividades descritas.

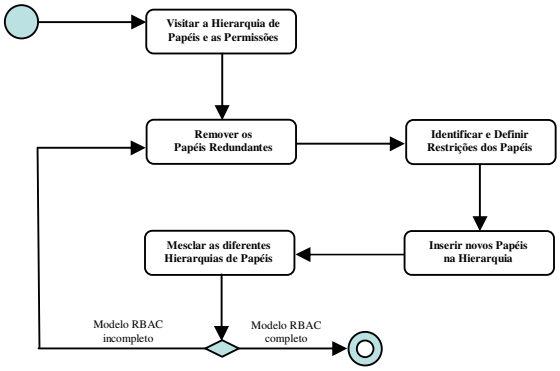

Figura 9: Sub-Processo de Definição do Modelo RBAC

3.2. Modelo de Inter-Relações

A Figura 10 mostra as inter-relações entre os documentos que são produzidos durante o Jogo do Planejamento XP integrado a Engenharia de Papéis:

- *Role-related User stories*: compreendem todas as estórias de usuários relacionadas aos papéis criados e servem como base para a integração.
- Cartões de Testes de Aceitação: compreendem todos os cartões de testes de aceitação relacionados às *Role-related User stories*.
- Catálogo de Permissões: consiste em todas as permissões identificadas num sistema. As permissões são derivadas diretamente dos cartões de testes de aceitação associados às *Role-related User stories*.

- Catálogo de Restrições: contém as restrições que precisam ser aplicadas às permissões.
- Tarefas: descrevem todas as tarefas que são executadas por certos usuários de um sistema. Cada tarefa consiste de uma relação com uma ou várias *Role-related User stories*.
- Papéis: consistem no agrupamento de tarefas que definem as funções que um usuário desempenha no sistema.
- Modelo RBAC: é o resultado final do processo da engenharia de papéis e consiste de todos os papéis do sistema organizados em uma ou mais hierarquias de papéis.

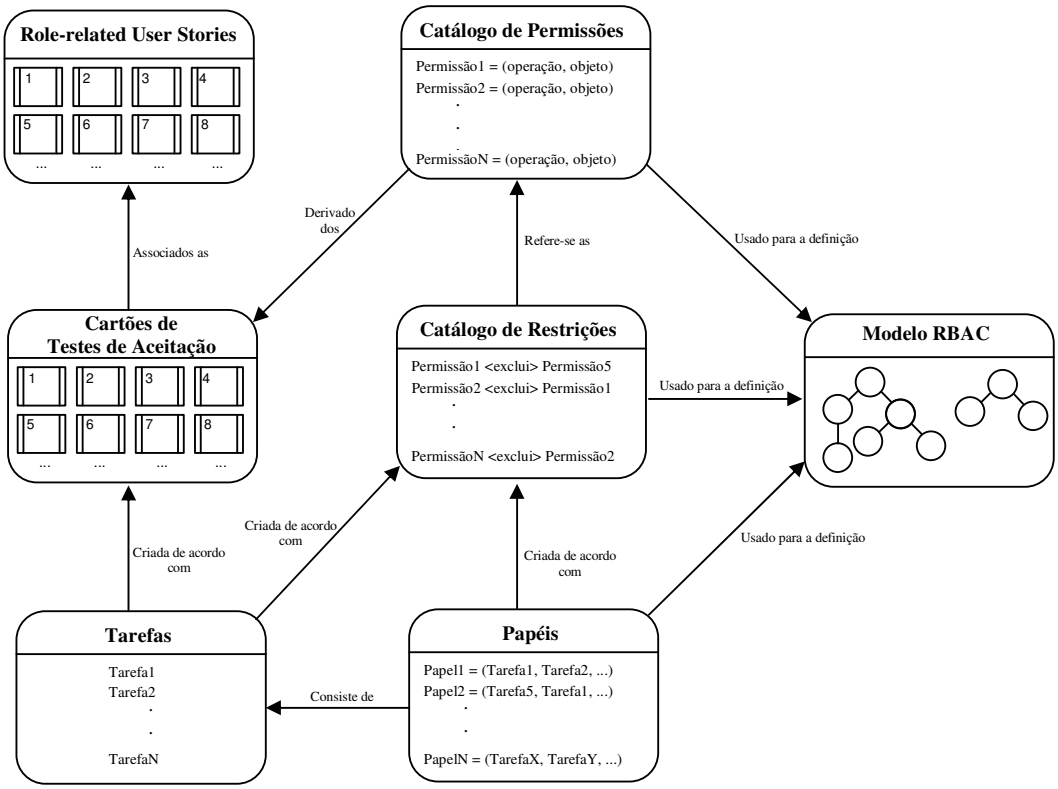

Figura 10: Inter-relações entre os documentos produzidos no Jogo do Planejamento integrado à Engenharia de Papéis

4. Discussão e Conclusão

A construção de uma proposta formalizando uma metodologia para a concepção de política de segurança para o controle de acesso baseado em papéis é o principal objetivo deste trabalho. Alcançou-se o objetivo à medida que fizemos a integração entre a engenharia de papéis e o jogo do planejamento *eXtreme Programming*, que é um dos métodos ágeis mais aceitos pelo mercado. Fazer um trabalho que integrasse estas duas áreas foi um desafio à parte, pois na literatura levantada não encontramos trabalhos científicos que abordassem explicitamente tais aspectos, ou seja, uma proposta metodológica que utilizasse especificamente um processo de desenvolvimento ágil e a engenharia de papéis de forma integrada. Também não encontramos trabalhos que propusessem a integração da engenharia de papéis aos processos de desenvolvimento

tradicionais. Dessa forma, uma análise comparativa entre o nosso trabalho e outros que utilizassem uma integração entre a engenharia de papéis e o processo de desenvolvimento de software, seja ele, ágil ou tradicional, não pôde ser feita.

Encontramos, porém, trabalhos na área de segurança que tratam o assunto de forma mais abrangente e não especificamente do controle de acesso baseado em papéis. Wäyrynen et al. (2004) apresentam um estudo importante para avaliar se a engenharia da segurança e o *eXtreme Programming* poderiam caminhar juntos. Eles analisaram o XP do ponto de vista de dois padrões da engenharia da segurança, o Systems Security Engineering-Capability Maturity Model (SSE-CMM, 2007) e o Common Criteria – CC (Commom Criteria, 2007), chegando à conclusão que o XP está alinhado à engenharia de segurança. Siponen et al. (2005) apresentam um exemplo de como integrar as técnicas de segurança aos métodos ágeis de uma forma genérica, eles também ilustram esta solução através de um método de desenvolvimento ágil batizado de Feature Driven Development (FDD). Peeters (2005) apresenta uma extensão das *User stories* do jogo do planejamento XP para suportar requisitos da engenharia de segurança, chamando esta extensão de *Abuser stories*. Boström et al. (2006) usando o conceito das *Abuser stories* desenvolvido por Peeters, apresentam uma extensão completa do jogo do planejamento XP para suportar os requisitos da engenharia de segurança. Seguindo as idéias de Peters e Boström et al. construímos as nossas *Role-related User stories*, agora com a finalidade específica de levantar requisitos para o efetivo controle de acesso baseado em papéis.

Na área de engenharia de papéis, consideramos adequado usar alguns processos propostos por Newman e Strembeck por estes defenderem, tanto quanto o *eXtreme Programming*, ciclos de vida curtos e iterativos que forçam que a atividade de levantamento de papéis ocorra várias vezes no processo. O trabalho por eles apresentado é bem completo, porém não apresenta nenhuma integração ao processo de desenvolvimento de software.

Estamos atualmente trabalhando na validação de nossa proposta na prática, para determinar em primeiro lugar sua eficácia e em segundo lugar o quanto a engenharia de papéis é afetada pela abordagem ágil. Desta maneira, poderemos encontrar os pontos de conflito entre os métodos ágeis, especialmente o XP, método escolhido para a nossa proposta, e a engenharia de papéis. Acreditamos que a identificação destes pontos de conflitos de forma clara, será de grande importância na tentativa de fazer melhorias à nossa proposta, sendo este um novo trabalho a ser desenvolvido.

Referências

ANSI/INCITS 359-2004. "Information Technology: Role Based Access Control. InterNational Committee for Information Technology Standards", 56 p. Fevereiro de 2004.

Astels, D.; Miller, G.; Novak, M. "*eXtreme Programming* – Guia Prático". Rio de Janeiro, RJ: Campus, 2002.

Boström, G.; Wäyrynen, J.; Bodén, M. "Extending XP Practices to Support Security Requirements Engineering". In SESS'06, Shangai, China, 20-21 de Maio de 2006.

Commom Criteria. Disponível em: <http://www.commoncriteriaportal.org>. Acesso em: 20 de setembro de 2007.

Coyne, E. "Role Engineering". In Proceedings of 1st ACM Workshop on Role-Based Access Control, Gaithersburg, MD, Novembro de 1995.

Epstein, P.; Shandu, R. "Engineering of role/permission assignment". In Proceedings of

17th Annual Computer Security Application Conference, New Orleans, LA. Dezembro de 2001.

EXtreme Programming: A gentle introduction, 17 Febuary 2007. Produced by Don Wells. Disponível em: <www.extremeprogramming.org>. Acesso em: 20 de setembro de 2007.

Ferraiolo, D. F., Sandhu, R., Gavrila, S., Kuhn, D. R. e Chandramouli, R. "Proposed NIST Standard for Role-Based Access Control", ACM Transactions on Information and System Security, v. 4, n. 3, p. 224-274, Agosto de 2001.

Fowler M. "The New Methology". Dezembro de 2005.

Howard, J. D. "An Analysis of Security Incidents on the Internet: 1989-1995", Carnegie Mellon University, PhD Thesis, 1997.

Joshi, J. B. D.; Aref, W. G.; Ghafoor, A.; Spafford, E. H. "Security models for web-based applications". Communications of the ACM, v. 44, n. 2, p. 38-44. Fevereiro de 2001.

Neumann, G.; Strembeck M. "A scenario-driven role engineering process for functional RBAC roles". In Proceedings of 7th ACM Symposium Control Models and Technologies, Monterey, CA. Junho de 2002.

Peeters J. "Agile Security Requirements Engineering". Presented at the Symposium on Requirements Engineering for Information Security, 2005.

Roeckle, H.; Schimpf, G.; Weidinger, R. "Processes-oriented approach for role-finding to implement role-based security administration in a large industrial organization". In Proceedings of 5th ACM Workshop on Role-Based Access Control, Berlin, Alemanha, 26-27 de Julho de 2000.

Shandu, S.; Ahn G. J. "Role-based authorization constraints specification". ACM Transactions on Information and System Security. Novembro de 2000.

Shandu, S.; Coyne E. J.; Feinstein H. L.; Youman C. E. "Role-based access control models". IEEE Computer, pág. 38-44, fevereiro de 1996.

Shin, D.; Ahn, G. J.; Cho, S.; Jin S. "On Modelling System-centric Information for Role Engineering". ACM. Junho de 2003.

Siponen, M.; Baskerville, R.; Kuivalainem, T. "Integrating Security into Agile Development Methods". In Proceedings of the 38th Hawaii International Conference on System Sciences. 2005.

SSE-CMM: Systems Security Engineering - Capability Maturity Model. Disponível em: <http://www.sse-cmm.org/index.html>. Acesso em: 20 de setembro de 2007

Thomsen, D.; O´Brien, D.; Bogle, J. "Role Based access control framework for network enterprises". In Proceedings of 14th Annual Computer Security Application Conference, pages 50-58, Scotscale, AZ, 7-11 de Dezembro de 1998.

Wäyrynen, J.; Bodén, M.; Boström, G. "Security Engineering and *eXtreme Programming*: an Impossible marriage?", XP/Agile Universe 2004, Berlin: Springer-Verlag, 2004, pp. 117-128.