# Tailoring Rational Unified Process to Contemplate the SSE-CMM

Rosana Wagner[a]

*Instituto Federal Farroupilha*
*Alegrete, Brasil*

Lisandra Manzoni Fontoura[b]

*Centro de Tecnologia*
*Universidade Federal de Santa Maria*
*Santa Maria, Brasil*

Raul Ceretta Nunes[c]

*Centro de Tecnologia*
*Universidade Federal de Santa Maria*
*Santa Maria, Brasil*

**Abstract**

This paper describes an extension proposal of the process framework named Rational Unified Process (RUP) so that it contemplates the security practices proposed by the System Security Engineering Capability Maturity Model (SSE-CMM). It was possible to check that several process areas proposed by the SSE-CMM are not contemplated by the RUP through the comparison between those process model. We believe that the incorporation of security, based on the SSE-CMM, to the RUP is important so that the security aspects are considered from the beginning and throughout the whole life cycle of the software development, therefore avoiding that the software that is given contains security vulnerabilities. Having that considered, this paper proposes the extension of the Unified Process by means of the inclusion of a new discipline in the RUP which seeks to satisfy security requirements as it is described by the SSE-CMM model (standard ISO/IEC 21827) in a way that the security is integrated into all the software development phases.

*Keywords*: Software Process, Security Management, SSE-CMM, RUP

[a] Email: rosanawagner@gmail.com
[b] Email: lisandramf@gmail.com
[c] Email: raul.cerettanunes@gmail.com
We want to express our gratitude to CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) for the financial support

# 1. Introduction

Nowadays the maturity in the software development is achieved by using good software engineering practices. Software development processes must guide the developer in an organized way and indicate the usage of methods, techniques and tools that allow the elaboration of high quality software. With the intent of obtaining such quality, it is necessary to consider security requirements throughout all the phases of the software development, not only in the final stages (testing and homologation) [1] [2].

According to a study conducted by [3], the stealing of data and violation of cybernetic crimes might have cost more than $ 1 trillion for the companies in 2008 due to the loss of intellectual propriety and the expenses with the repairing of the damages. These data are worrying and if they could have been predicted, probably, they would be avoided by implementation of some standard and security models, such as: ISO 27001, ISO 27002, SSE-CMM, among others [2].

The SSE-CMM [4] model is focused on the processes that are used for achieving the Information Security, more specifically on the maturity of these processes. The SSE-CMM model was a result from the investigation upon the necessity of a Capability Maturity Model (CMM) that is specialized for handling with security engineering.

The Rational Unified Process (RUP) is a software engineering process. It provides a disciplined approach to assign tasks and responsibilities in a development organization. Its goal is to produce, within a predictable schedule and budget, high-quality software that meet the needs of its users [5].The RUP does not propose specific activities to handle security issues; it limits itself to describing that, in the "Detail Requirements" activity, it is necessary to consider as non-functional requirements the system security requirements and that, in the "Review Architecture" activity, it is necessary to evaluate whether the architecture contemplates the defined security requirements or not.

Considering that the goal of the SSE-CMM is not specifying a standard process to be used by the organization but the intention is that the organization can use processes by tailoring them to the recommendations of the SSE-CMM [4], and that the Rational Unified Process is a software process model widely used nowadays, this paper proposes an extension of the RUP so that it can fulfill the security practices of the SSE-CMM by tailoring it for the secure software development, according to a security model that has been successfully used in several organizations.

The RUP which is extended for security requirements may be used by the organizations that are looking for developing reliable systems, guaranteeing the security in all phases so that it does not exist in the end of the redone project for integrating the security to the system. The organization may use the RUP extension proposed by this article or tailor their software processes consistent with the activities that are here detailed which are responsible for the implementation of the security according to the practices of the SSE-CMM.

The tailoring of the RUP consists in the insertion of a new support discipline named "Software Security" which describes a series of activities, tasks, artifacts and roles that are defined according to the specification of the SSE-CMM Model [4].

The article is outlined as follow: Sections 2 and 3 briefly describe the Rational Unified Process (RUP) and the SSE-CMM, respectively, by presenting their structures and organizations. Section 4 describes the proposal for extension of the RUP and the obtained results. Section 5 mentions some related work; and the final considerations and future work are described in Section 6.

## 2. Rational Unified Process

The Rational Unified Process is a software development process that describes the set of activities needed to transform user requirements into a software system. However, the Rational Unified Process is more than a single process; it is claimed to be a generic process framework that can be specialized for a large class of software systems, for different application areas, different types of organizations, different competence levels, and different project sizes

The RUP shows how a software development team can apply approaches that are commercially proved to be software development [7]. RUP is focused on six key principles of the software engineering which are known as "Best Practices". These principles constitute the fundament of the RUP, which are: adapt the process; balance stakeholders' priorities; collaborate across teams; demonstrate value iteratively; elevate the abstraction level; focus continuous on quality [6].

The iterative development on the RUP promotes and organizes the software development in four phases, each one composed of one or more iterations. Disciplines play an important role in designing the iterations carried out within each phase. Discipline is defined as a categorization of activities based on similarity of concerns and cooperation of work effort. Each iteration is formed by a set of disciplines whose emphasis vary depending on the phase in which the project is.

Each phase has well-defined goals which are verified at the end of the phase in the so-called milestones. The *Initiation* phase is mainly focused on defining lifecycle objectives of the project (project's scope); the *Elaboration* phase is about planning the project, specifying resources, defining and validating the architecture; the *Construction* phase is for constructing the product; and the *Transition* one is for implanting the software [7].

A discipline is a collection of activities that are related to an area of concentration or field of study. Each activity is decomposed into subactivities or tasks. The RUP proposes nine disciplines which are divided into six disciplines that are directly related to software engineering, named core disciplines, and three support disciplines. The core disciplines are: Business Modeling, Requirements, Analysis and Design, Implementation, Test and Deployment. The support disciplines are: Configuration and Change Management, Project Management and Environment [8].

The RUP uses three main elements for description of methods, which are: role, task and artifact [6]. The *role* defines the behavior and the responsibilities of an individual, or a set of individuals working in a group inside the context of a software organization. The *role* represents a job executed by individuals in a project and defines how they must do the job. A *task* is a unit of job that an individual, who plays a *role,* is in charge of executing. A *task* has a clear finality which is usually expressed in terms of creation or updating of some *artifact*, like a model, a class or a plan. *Tasks* have *artifacts* like entrance and exit. An *artifact* is a job product of the process.

## 3. System Security Engineering Capability Maturity Model (SSE-CMM)

The SSE-CMM model describes the essential characteristics that must exist in a security engineering process so that it is considered secure. The model is centered in the requirements for security implementation in a system or of a series of systems that are related to the domain of the Information Security Technology. Inside this domain the SSE-CMM is focused on the engineering that is used for achieving maturity in the development process [2].

The foundation practices for security of the SSE-CMM model are: PA01 – Administer Security Controls; PA02 – Assess Impact; PA03 – Assess Security Risks; PA04 – Assess Threats; PA05 – Assess Vulnerabilities; PA06 – Build Assurance Argument; PA07 – Coordinate Security; PA08 – Monitor Security Posture; PA09 – Provide Security Input; PA10 – Specify Security Needs; PA11 – Verify and Validate Security [4].

On the other hand, the implantation of the SSE-CMM for bettering the processes follows the IDEAL model of the SEI [4]: *Initiate* (setting for a successful improvement effort); *Diagnose* (determining where you are in relation to where you want to get); *Establish* (planning the details on how you will reach your destiny); *Act* (executing the job according to the plan); *Learn* (learning from experience and improving your skill).

The intention of the foundation practices and the improvement model is that an organization is able to use the SSE-CMM model to assess their processes' security, even if they use other orientation models of Information Security Technology [2], such as the RUP development process.

The SSE-CMM divides security engineering into three basic areas: risk, engineering and assurance [4]. The Risk area seeks to identify and prioritize risks that are associated to the products or systems development. The risks are assessed by examining threats and vulnerability probabilities besides considering the impact of an unwanted incident. This area covers these practices: PA04 – Assess Threats; PA05 – Assess Vulnerabilities; PA02 – Assess impact and PA03 – Assess Security Risks.

The Security engineering area is a process that goes through every step of the development, beginning in the system conception and moving forward in the project, implementation, testing, delivery, operation, maintenance, and discontinuity. The SSE-CMM emphasizes that the security engineers are part of a bigger group and need to coordinate their activities along with engineers from other disciplines. It helps evaluating that security is integrant part of bigger processes, not being a distinct separate activity. The difficulty in integrating this activity with the rest of the engineering process is that the solutions cannot be selected only by taking security into consideration; there is a large variety of other considerations, including cost, performance, technical risks and usage facility. This area covers these practices: PA10 – Specify Security Needs; PA09 – Promote Security Information; PA01 – Administer Security Controls; PA08 – Monitor Security Posture; PA07 – Coordinate Security;

Finally, the Assurance area seeks to certify that the implemented solutions are reliable. The assurance may be seen as confidence that the protections will work properly. This confidence comes from the properties of correctness and effectiveness. Correctness is the property that the safeguards, as designed, implement the requirements. Effectiveness is the property that the safeguards provide security adequate to meet the customer's security needs. This area covers the following

practices: PA11 – Verify and Validate security, PA06 – Build Assurance Arguments and several other PAs that together make assurance evidences.
The goals of the process areas (PAs) are achieved through the applying of the base practices (BPs) which are defined in the SSE-CMM model, using examples and concepts that are to be used in order to facilitate the model application.

## 4. Extending the RUP to Comply to the SSE-CMM Model

The RUP is widely used in major projects, also it is possible to be tailored to minor and medium ones, or customized to fulfill the specific needs of a given software project [6]. A previous work of one of the authors of this paper describes an extension for the RUP to comply with Capability Maturity Model for Software (CMM) [9] levels 2 and 3. This work uses the SSE-CMM to tailor the RUP according to the security recommendations of this model.

Considering that the SSE-CMM Model describes several recommendations that must be followed for the secure software development and that disciplines in the RUP seek to group activity collections related to a concentration area, it is opted to group the proposed activities in order to add security to the RUP in one single discipline, named "Software Security", according to Figure 1.

Another alternative would be to set activities related to security in the existing disciplines in the Unified Process, for example, "Specify Security Needs" could be inserted to the Requirements discipline. Several disciplines would have their activity diagrams altered, which makes difficult for the understanding and implementation of the process. Another advantage of having a separate discipline for dealing with security issues is the facility that organizations will have if they wish to extend their process based on this work.

This discipline has been elaborated according to the orientations for tailoring of the RUP [7], and it must be executed in all phases but with more intensity in the initiation and elaboration phase. This is a support discipline because it concerns security management within the project.

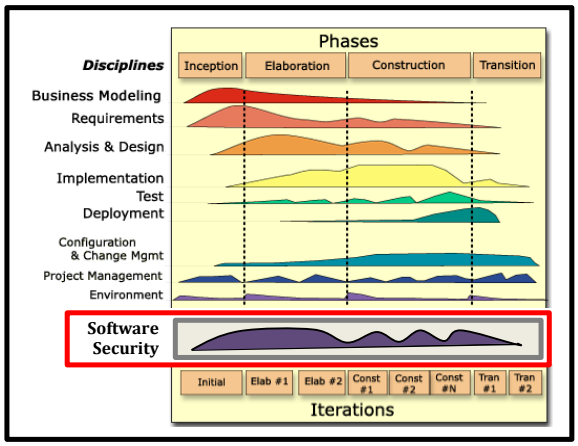


Figure 1. Definition of a new Discipline: Software Security

## 4.1. SECURITY DISCIPLINE DETAILING

After a detailed analysis of the SSE-CMM Model, a set of activities with the intent to contemplate the security according to this model was defined. These activities are organized in a UML activity diagram shown in Figure 2.

Considering that in the RUP the activities refer to a task grouping. Each activity of this diagram is described in separate diagrams as a task set that composes it, the roles that are responsible for executing each task and the input and output artifacts. This description format is the standard that is adopted by the RUP.

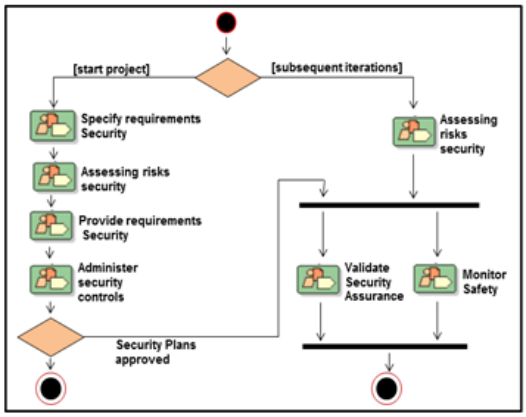


Figure 2: UML activity diagram with the security activities

According to *PA10 (Specify Security Needs)*, it is necessary that the organization explicitly specifies their security needs related to the system (security requirements). In order to be possible to understand and identify those security necessities it is necessary to check the *legal, political and organizational Requirements* that are related to the software security. The tasks shown in Figure 3 have been proposed in order to accomplish this PA. The *Understand Security Needs* task has the goal of common understanding upon the security needs under the customer and the stakeholders, which are documented in the *Security Requirements* artifact. The *Identify Security Context* task seeks to identify requirements, through the *Software Requirements Specification* document, that are necessary in the security context and to document in the *Security Requirements* artifact. The *Review Security Requirements* task is necessary for formally checking whether the security requirements are correctly documented in the *Security Requirements* artifact or not. The result of the review is documented in the *Record Review*.

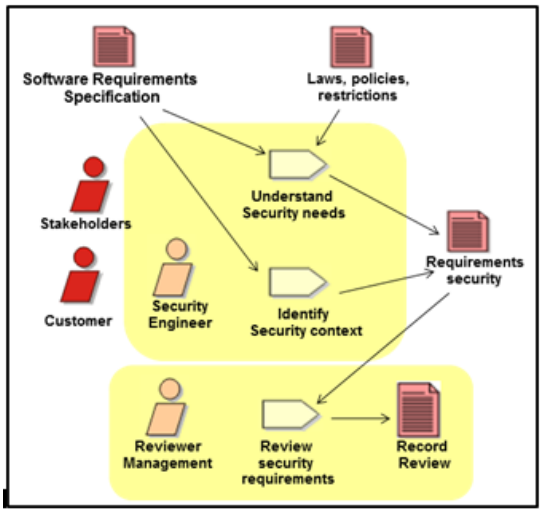


Figure 3. Specify Security Needs

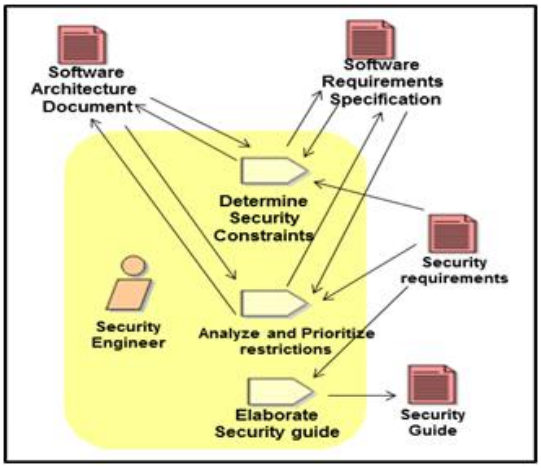


Figure 4. Provide safety requirements

According to *PA09 (Provide Security Input),* it is necessary to provide the system architects, designers, developers and users with information about the security that are necessary for assurance the implementation of the security requirements defined by the client, the stakeholders and the security engineer. In order to reach this result it is necessary that the security constraints, which are defined previously, are determined, analyzed and prioritized. The tasks shown in Figure 4 have been proposed in order to accomplish this PA. The *Determine Security Constraints* task has the objective of determining which actions must be done so that the security constraints are answered and documenting them in the *Software Architecture Document* and *Software Requirements specification.* The *Analyze and prioritize*

*restrictions* task has the objective of analyzing the restrictions that are determined according to security requirements and prioritizing them. The *Elaborate security guide* task has the objective of elaborating a S*ecurity guide* that will be used by the group in order to make decisions on architecture, project and implementation.

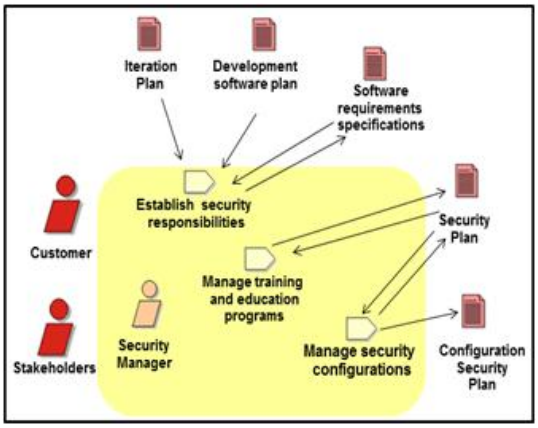


Figure 5: Administer security controls

According to *PA01 (Administrate security controls),* it is necessary to guarantee that the expected security for the system has been integrated and will be fulfilled when the system is in operation. The tasks shown in Figure 5 have been proposed in order to accomplish this PA. The *Establish security responsibilities* task has the objective of elaborating the *Security plan* considering the requirements that are described in the *Software requirements specification* artifact which must be followed during the software development. The *Manage trainings and education programs* task must identify trainings and education programs that are necessary and include specific sections in the *security plan* based on its responsibilities. The *Manage security configurations* task has the objective of elaborating the security *Configuration plan* by documenting the procedures that will be used for security configuration managing, such as software updating registries, version registries, and change follow-ups. In order to answer to the SSE-CMM model it is also necessary that those activities are coordinated with the purpose of guaranteeing that all parts are aware of and involved with the security engineering activities. This task includes an open communication among all members of the project.

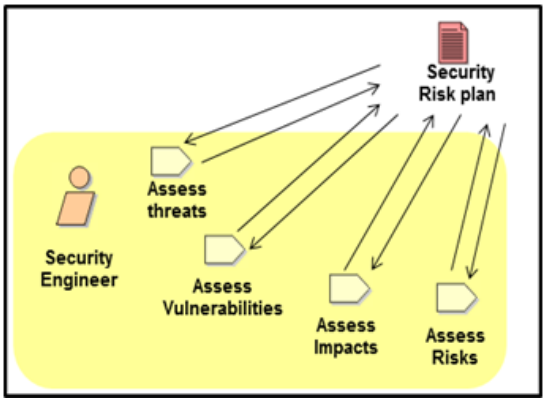


Figure 6. Assess security risks

According to *PA02 (Assess Impact), PA03 (Assess Risk), PA04 (Assess Threat) and PA05 (Assess Vulnerability)*, it is necessary to assess risks related to threats, vulnerabilities and impacts thus generating a security risk plan. The tasks shown in Figure 6 have been proposed in order to accomplish this PA. The *Assess Threats* task has the objective of identifying threats to the system, their properties and characteristics and documenting in the *Security Risk Plan*. The *Assess Vulnerabilities* task has the objective of identifying the vulnerabilities that may occur in the system, including system evaluation analysis, specific vulnerabilities definition and the promoting of a guarantee upon the whole system. The *Assess Impacts* task has the goal of assessing the impact occurrence probability. The *Assess Risks* task has the objective of identifying the risk exposure in a defined environment and prioritizing them. These four tasks document in the *Security Risk Plan* the measures that must be taken in order not to occur system threats, vulnerabilities and impacts.

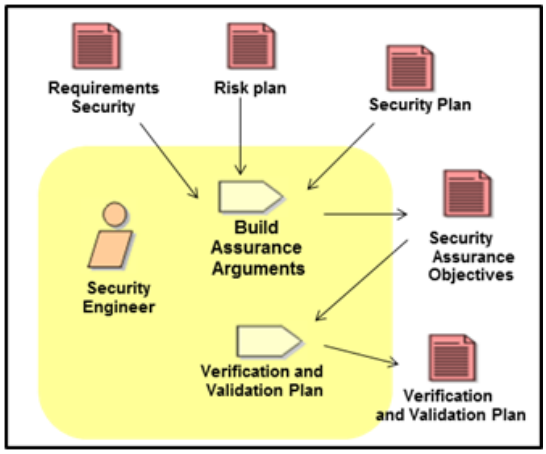


Figure 7: Validate Security Assurance

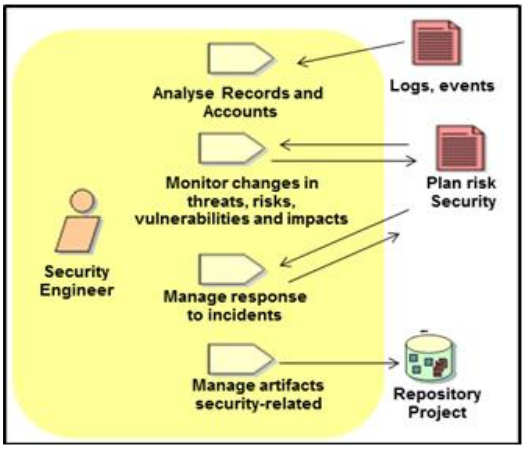


Figure 8. Security Monitor

According to *PA11 (Verify and Validate Security),* it is necessary to assurance that solutions have been verified and validated towards security. The tasks shown in Figure 7 have been proposed in order to accomplish this PA. The *Build Assurance Arguments* task elaborates a document with the *Security Assurance Objectives* where the expected priorities and objectives with the implementation of the security assurance are detailed. The *Verification and Validation plan* task has the objective of developing a verifying and validating plan to be used during the system development. This plan must be used to check whether the security requirements have been found and the customer's operational security needs have been contemplated or not.

According to *PA08 (Monitor Security Posture)*, it is necessary to ensure that all breaches of, attempted breaches if, or mistakes that could potentially lead to a breach of security are identified and reported. The tasks shown in Figure 8 have been proposed in order to accomplish this PA. The *Analyze Records and Accounts* task has the objective of checking logs and events in order to identify incidents that may be considered as risks. The *Monitor changes in threats, vulnerabilities and impacts* task has the objective of constant monitoring in changes that may occur in relation to threats, risks, vulnerabilities and impacts. The *Manage Response to incidents* task has the objective of providing responses to the changes that occur in the new incident. The *Manage artifacts security-related* task has the objective of updating the project repository with new security information.

A case study was elaborated with the intent to qualitatively evaluate the difference between both models, RUP and Tailored RUP. The RUP model was used on the first case study, and the tailored RUP according to the SSE-CMM model was used on the second one. Analysis and comments on the presented case study are shown at the end of this section.

This case study shows an Information System in which the security is a critical aspect. ReceitaNet will be used as an example system.

## 5. TAILORED RUP CASE STUDY

In order to describe how to use the tailored RUP model to contemplate the SSE-CMM security, we will adopt a client personal data register system for the annual Income Tax (IRS) declaration in an accounting organization.

It is necessary that the professional uses common sense and ethics during the data register but, once the data are in the system and the organization has access to all the accounts, values, payment sheet and receipts from their clients, the responsibility shifts to the company system.

The data are not only registered, but also sent to the government in this system. The indicated security for the development must come to action at the moment of sending the data through online transactions.

During the information transference operation several security failures may occur, caused by systems vulnerabilities. According to what Robertson [10] assures in his thesis summary, the World Wide Web involves from a service for interconnected systems such as static documents for what there is a powerful, versatile, and widely democratic service platform to the delivery of applications and the information dissemination. Unfortunately, along with the web growth came also a great increase on the number of impacts and security incidents. The magnitude of this problem increased much of the interest with what the security community develops these search mechanisms [11].

Bearing that in mind, the Software Security discipline that is proposed in this paper consists in the making of a series of activities, shown in Figure 2, that seek to avoid security failures. In the following, a brief description of the activity in relation to the specific case study.

The activity named "Specify Security Needs" consists in identifying the possible threats that may happen, checking problems that have previously occurred in relation to the security in this kind of system. It is taken into consideration that the information is the most important active and the confidentiality, availability, integrity, authenticity and non-repudiation of the users of these services must be assured [11]. For the Income Tax System it is possible to cite hacker's attacks to the IRS's servers which may put clients' data at risk. For example, it is possible to cite the recent attack to Google's servers when the password system of the company was violated [12]. After identifying the security requirements, they are documented and later reviewed for assurance that they fulfill every stakeholder's needs. It is possible to come to an agreement upon a series of security definitions based on standard as proposed in the ISO/IEC 27001.

The "Assess Security Risks" activity consists in identifying threats, vulnerabilities, impacts and risks that may occur in this kind of system. Examples of security risks associated to the Income Tax System include: Cross-site scripting (XSS) which are among the most prevalent attacks amid the web applications [13]; Cross-site request forgery (CSRF) [14]; HTTP header injection which is a class of vulnerability that involves several variations [10]; SQL injection which is not a specific kind of attack for the web [14]. In his thesis, Robertson (2009) affirms that clients and Web servers have been vulnerable to command injection attacks. In particular, the web applications that execute external programs during the processing of the order without properly checking the arguments of the given client have proved to be popular attack vectors. These vulnerabilities are considered major ones since arbitrary actions may be executed.

The "Provide Security Requirements" activity consists in altering the "Software Architecture Document" and "Software Requirements Specification" artifacts in order to incorporate the security requirements that are identified for the project. Examples of security requirements that may be incorporated to the Income Tax System project include: checking access permissions to the system's options according to the user's login; checking if the connection has been established on the correct server at the moment of data transmission; checking if the computer with the

web access that will send to the organization server is safe, etc.. In [15], after analyzing bad use cases, it was possible to identify several kinds of potential threats, like: unauthorized information disclosure; unauthorized data alteration; unauthorized information unavailability; Phishing.

The "Administer Security Controls" activity establishes the responsibilities for the making of the activities related to security, as well as it identifies the training needs according to the responsibilities and predicts the elaboration of a plan for security configuration management. For the Income Tax System these artifacts must be created according to the organizational context.

The "Validate Security Assurance" activity consists in elaborating and prioritizing activities that must be done in order to validate the implemented security mechanisms. These assurance requirements may be obtained through the EAL (Evaluation Assurance Level) which is established according to the interested parts, the stakeholders. For the Income Tax System it is possible to hire hackers, people that do not know the system in order to try to break it, identifying vulnerabilities so that those can be corrected before the system is implemented.

The "Security Monitor" activity consists in analyzing through logs that are available to the violation attempts that have occurred and response to these attacks in an adequate way. In this case, the fraud attempt logs that have been made must be taken into consideration and check if none of those attempts will possibly result into a future improper access to the system.

Being an iterative and incremental process, several iterations will be done until it is possible to develop the complete software. At the beginning of each iteration, it is necessary to verify if the security needs are still valid for the current iteration and if no other necessity is identified. Improvement in relation to the process that is used in the previous iteration may also occur as a result of the evaluation at the end of the iteration.

## 5.1 ANALYSIS AND COMMENTS

When developing the data transfer system of the IRS Tax Income it is necessary to consider activities, tasks and roles, that is, the whole recommended context by the RUP. The main goal of the development aims at completing the implementation of a system that has security requirements. However, using the RUP model is not about security as we have referred to through the existing bibliography, which leads security to be approached by means of the other disciplines and phases that the models present. Yet, in the tailored RUP it is possible to verify security as main issue the same way the other disciplines are approached.

We understand that our case study is quite limited and that this part is important and it needs elaboration. However, in order to be conclusive, we know that one or two case studies are not going to be enough for the validation of the model but many case studies with development in different group sizes must be considered.

## 6. RELATED WORKS

Some works propose RUP extension in order to fulfill security requirements. Two works specially are cited.

Paes e Hirata (2008) proposes a RUP extension in order to contemplate security. These authors also propose a RUP extension in order to contemplate the development of fault tolerant software [17].

Tovar et al (2006) describe an analysis of how the management processes COBIT and ITIL and the security model SSE-CMM may contribute for the development of

secure software, also highlighting overtaking aspects between these models in relation to software security. The main focus of this article is in the comparison between the security aspects mentioned in the presented models.
Even though the cited works are related to security, they are not based on a security model as the present work is. It is considered important that models and standard which have been already approved are used for the elaboration of processes that seek to develop secure software.

Considering that security, once it is proposed through a consistent model, will have results that are more objective and focused in the proposed model, the user will recognize which security level will be contemplated in the system. The tasks that are proposed in this work contemplate to level two of the SSE-CMM because they focus on the definition problems, planning and development in terms of project.

## 7. CONCLUSIONS AND FUTURE WORKS

The security in which the system development is based on has seemed to be more and more important for the organization, the developers and the users.

The RUP does not describe activities, roles and artifacts for the implementation of security in a satisfactory way. Having that considered, this work seeks to contribute to secure software development by means of the definition of a process following practices recommended by a model that is much used by the organizations.

This way, the present work submits a model of the tailored RUP with an extra discipline which contemplates security according to the SSE-CMM model. It is very important that the security is integrated since the beginning of the project, going through the initiation, elaboration, construction and transition phases.

The discipline that is proposed to contemplate the security requirements in the RUP model has been described by means of an activity diagram which provides a wider understanding and, later, the roles, tasks and artifacts have been described in order to answer to the security that was demanded by the SSE-CMM model, according to each activity to be realized.

The process that was proposed is useful for organizations that use the RUP and that wish to improve their processes by inserting information security related practices so that the incorporation of the security requirements to the software is realized during the development of the same. Organizations that have their own software processes and aim at the development of reliable software may take the activities proposed in this work as a basis by incorporating them into the organizational process.

Future works include the elaboration of a methodology for the tailoring of processes based on security patterns and according to specific requirements of each project, also the execution of the tailored processes using Business Process Management Systems (BPMS).

## References

[1] Paes, C. and Hirata, C. (2007) "RUP *extension for the development of secure systems".* In: Proceedings of the International Conference on Information Technology, p.643-652, Washington, USA.

[2]Tovar, E., Carrillo J., Veja V. and Gasca G. (2006) "*Desarrollo de productos de Software seguros en sintonía con los Modelos SSE-CMM, COBIT e ITIL*". In: Revista de Procesos y Métricas de las tecnologías de la información. Universidad Católica del Norte - Universidad, v.3 n.2, p.60-69 Madrid, Spain.

[3] Mcafee. (2009) "*Cybercrime cost $1 trillion last year, study*". In: ZDNet News & Blogs/Technology News. Available at: <http://news.zdnet.com/2100-9595_22-264762.html 1>1

[4]SSE-CMM. (2003) "Systems Security Engineering-Capability Maturity Model Group (SSE-CMM) – Model Description Document". Version 3.0.

[5]Rational Software Corporation, (1998) "Rational Unified Process Best Practices for Software Development Teams". In: IBM Rational Software. Available at: <http://www.rational.com/media/whitepapers/rup_bestpractices.pdf.> .

[6]Shuja, A. (2008) "Welcome to the IBM Rational Unified Process and Certification". In: IBM Rational Software. Available at:<http://www.ibmpressbooks.com/bookstore/product.asp?isbn=0131562924 >

[7] Kruchten P. (2000) "*The Rational Unified Process – An Introduction*". Ed., Reading, Mass.: Addison-Wesley.

[8] IBM Corporation. (2007) "IBM Rational Unified Process" v7.0.

[9] Manzoni, Lisandra V., Price, Roberto T. (2003) "Identifying Extensions Required by RUP (Rational Unified Process) to Comply with CMM (Capability Maturity Model) Levels 2 and 3". In: IEEE Transactions on Software Engineering, v. 29, n. 2, p.181-192, New York, USA.

[10]Robertson K. W., (2009) "Detecting and Preventing Attacks Against Web Applications". Doctoral thesis. University of Califórnia, Santa Barbara, USA.

[11]Cardoso et al. (2009) "Um Modelo de Controle Formal para o gerenciamento de riscos de processo em fábricas de *software*". In: Proceedings of the IX Brazilian Symposium on Information Security and Computer Systems. Campinas, Brazil.

[12]Gorman S. and Vascellaro J.E. (2010) "Google Attack Linked To Asian Hackers". Available at: http://online.wsj.com/article/SB10001424052748704751304575080362745174130.html

[13]Kiezun A., Guo P., Jayaraman K., Ernst M., (2009) "Automatic Creation of SQL Injection and Cross-Site Scripting Attacks" In: Proceedings of the 31st International Conference on Software Engineering, v.31, p.199-209. Washington, USA.

[14] Alexenko T., Jenne M., Deb Roy S., Zeng W., (2010) "Cross-Site Request Forgery: Attack and Defense". In: 7th IEEE conference on Consumer Communications and Networking Conference. V.7 p. 342-343. Las Vegas, USA.

[15]Mellado D., Fernández-Medina E., Piattini M., (2007) "Un Proceso de Ingeniería de Requisitos de Seguridad en la Práctica" In: IEEE Latin America Transactions. V.5, n.4 p. 211 – 217.

[16] Paes, C. E. B., Hirata, C. M. and Yano, E. T. (2008) "*Extending RUP to Develop Fault Tolerant Software*". In: ACM Symposium On Applied Computing, Fortaleza, Brazil.