مجلة العلوم البحتة والتطبيقية
Journal of Pure & Applied Sciences
www.Suj.sebhau.edu.ly ISSN 2521-9200
Received 21/02/2020 Revised 07/08/2020 Published online 05/10/2020

A Hybrid SV-SCRUM Model: Integrating SCRUM in SV-Model

*Ala A Abdulrazeg[a] , Salim Ali Abdulrrzaiq[b]
[a]Department of Computer Science, College of Education/ Omar Al-Mukhtar University, Libya
[b]Department of Computer Engineering, College of Engineering/ Omar Al-Mukhtar University, Libya
* Corresponding author: ala.alaker@gmail.com

Abstract Agile models are widely used in the software industry due to their unique features, such as quick response to requirement changes, reduced documentation and agility. SCRUM is one of the most useful agile models that offer a unique management technique to reduce the risk of rework, delay, and scheduling for a good quality product. However, the SCRUM model has certain limitations, such as lacks in engineering practice, fully depends upon customer, lacks in structured reviews. SCRUM is also in direct conflict with security engineering activity and techniques. SV-Model is a secure plan-driven method that encourages a proactive definition of security requirements. The SV-model is criticized of being inflexible and slow to respond to rapidly changing requirements. Although both models SV-Model and SCRUM contain good features and strengths but still there are improvement possibilities in these models. Keeping above mentioned context in view, this paper intends to propose a Hybrid SV-SCRUM Model by accommodating SCRUM management practices in the SV-Model to expedite product development. The biggest benefit of using SCRUM and SV-Model together is, SCRUM is an agile framework and SV-Model is a secure engineering methodology. This combination is primarily done to harness the qualities of the SV-model and reduce its weaknesses by incorporating the strengths of the SCRUM approach.
Keywords: Agile process, Hybrid development model, SCRUM, Security Requirements, SV-Model.

نموذج SV-SCRUM الهجين: دمج نموذج SV مع SCRUM

*علاء على عبدالرازق[1] و سالم على عبدالرازق[2]
[1] قسم الحاسوب–كلية التربية–جامعة عمر المختار، ليبيا
[2] قسم هندسة الحاسوب–كلية الهندسة–جامعة عمر المختار، ليبيا
*للمراسلة: ala.alaker@gmail.com

الملخص تُستخدم النماذج الرشيقة على نطاق واسع في مجال تطوير البرمجيات نظراً لميزاتها الفريدة مثل مواكبة تطوير وتغيير متطلبات المشروع و تقليل التوثيق و الخفة في الحركة. SCRUM تعتبر واحدة من أكثر النماذج الرشيقة التي تقدم تقنية لإدارة تطوير المنتجات التي تقلل من مخاطر إعادة العمل و التأخير و الجّدولة لتطوير منتج عالي الجودة. لكن نموذج SCRUM له مجموعة من نقاط الضعف، مثل عدم تقديم خطوات واضحة لتطوير البرمجيات و الإعتماد بشكل كبير على العميل و التضارب مع هندسة المتطلبات الأمنية. SV-Model هو نموذج تطوير برمجيات آمنه من خلال تعزيز مرحلة هندسة المتطلبات في النموذج بمجموعة من الخطوات الأمنية، تم انتقاد SV-Modelعلى انها غير مرنة وبطيئة في الإستجابة لتغيير متطلبات المشروع بسرعة، على الرغم من أن كلا النموذجين SV-Model و SCRUM يحتويان على نقاط قوة جيدة، لا تزال هناك إمكانية تحسين في هذه النماذج، وعليه، تقدم هذه الورقة كمقترح لنموذج الهجين باسم SV-SCRUM من خلال إضافة تقنيات إدارة SCRUM في نموذج SV-Model لتسريع تطوير المنتج، الفائدة الكبرى في دمج SCRUM و SV-Model ، هي أن SCRUM هو نموذج رشيق و SV-Model هو نموذج تطوير برمجيات آمنه، و هذا المزيج يحافظ على نقاط قوة SV-Model وتقليل نقاط ضعفه من خلال دمج نموذج SCRUM.
الكلمات المفتاحية: النموذج الرشيق، النموذج الهجين، سكرم، هندسة المتطلبات الأمنية، نموذج SV الأمن.

Introduction

In software engineering, there exist two main software development approaches: the plan-driven approach and the agile approach. Plan-driven approach include, besides others, the Waterfall model or the V model. These approaches are based on a linear structure, which means that only the previous phase be ultimately ended, and the next phase begins.

Their greatest strength based on accurate planning, up-front extensive documentation of the entire process and having robust design concepts before starting to code. However, these methods criticized of being inflexible and slow to respond to requirements change of customers' evolving needs [1]. The agile approach emerged to counter the weaknesses of plan-driven approach. Common approaches of agile are SCRUM, eXtreme Programming (XP), and Kanban. Agile development methods gain speed in delivering a product by depending on the on-site customer and presenting user requirements in the documentation form of user stories with no

detailed requirements and design documentation created [1]. Agile development methods are characterized by multiple incremental iterations that assist in the ability to respond to change.

Plan-driven methods in contrast with agile methods are more preferable in producing high-quality software and providing high assurance. Akrout et al. [2] explained that the short development iterations, minimal design upfront, direct communication, and gradual building of test cases of agile methods, it is difficult to assure the quality of the software under development. Therefore, in recent years several studies were conducted to integrate non-functional requirement engineering processes that define the quality attributes of software such as security with the plan-driven methods. Abdulrazeg et al. [3] proposed a Secure V-Model (SV-Model), an enhanced version of V-Model that cares for security requirements during the requirement phase. Like any plan-driven model, SV-model is criticized of being less active and slow to respond to requirements change of customers' evolving needs.

Due to a market pressure of delivering high-quality software on a short time, the focus has therefore been directed towards blending an agile approach into a plan-driven approach to create so-called Hybrid System Development Life Cycles (short: hybrid approaches) that leverage the benefits of both approaches [4]. In this paper, we proposed a hybrid secure development model by accommodating SCRUM management practices in the SV-Model to expedite product development. SCRUM is chosen to be blended in the SV-Model as an execution strategy to maintain iterative development and rapid customer feedback. The proposed model is named a Hybrid SV-SCRUM Model that leverages the benefits of both the SV-Model and the SCRUM. The Hybrid SV-SCRUM Model ultimately aims at simultaneously catering for the challenges of rapid requirements change phenomenon and the need for building a secure system.

SV-Model

The Secure V-Model (SV-Model) is a security-enhanced version of V-Model. The SV-Model incorporates a group of clear and stepwise security requirements engineering activities and techniques to ensure the level of security can be achieved by drawing attention to security in the requirements phase. In the SV-Model the left-hand side covers the development activities and the corresponding test preparation activities, while the right-hand side covers dynamic testing which verifies and validates the system under test by means of execution and observation of its behaviour, as illustrated in Figure (1).

The SV-Model starts off with eliciting, analyzing, specifying, and validating functional requirements (user requirements) and security requirements. The SV-Model makes the specified functional and security requirements useful at later stages of development, specifically during the testing stage by generating acceptance test cases, functional test cases, security test cases, and risk-based security test cases. Once the requirements specification is developed the project goes to architectural and detailed design. Integration test cases are created during architectural design to verify the ability of the system components to work together. Unit test cases are created during detailed design to test individual component. Once the implementation phase is complete, the parallel structure moves upward from the implementation phase, giving the model its distinct V shape. The upward ladder describes each of the testing steps that follow coding, starting with unit testing, integration testing, and ending with system functional and security testing, risk-based security testing and, user acceptance testing.

SCRUM

One of the most popular and widely adopted agile methods is SCRUM [8]. It provides a simple set of principles, working practices, and roles for teams to execute. The SCRUM is a general agile method focusing on managing iterative development rather than specific technical approach. SCRUM has proved to be a highly successful method for managing iterative and incremental software development projects [4] [6].

The heart of SCRUM is the teamwork. It relies on a self-organizing, self-managing, cross-functional team that is supported by a SCRUM Master and a Product Owner. The product owner represents the interest of various customers and responsible for creating and maintaining a prioritized list of product features, called a product backlog. SCRUM project development advance via a series of sprints which are typically 2-4 weeks in length with an initial planning phase and a final closure phase of a sprint review, as illustrated in Figure 2.

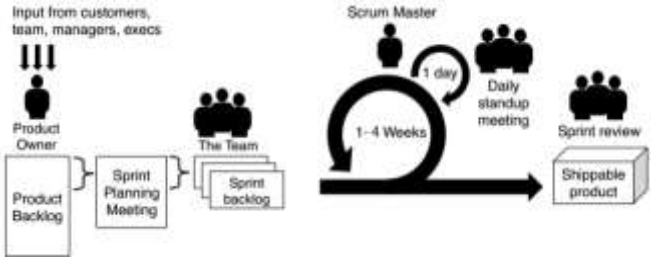

Figure 2 SCRUM development Process [15]

At the beginning of each Sprint, the Sprint Planning Meeting takes place. In the sprint planning meeting the product owner and the SCRUM team select items from the product backlog and commits to complete them by the end of the Sprint. Each workday, the team gathers briefly in a Daily Meeting to report the progress, and update simple charts that orient them to the work remaining. At the end of each sprint, the Sprint Review takes place. In the sprint review, the team demonstrates what they have built during the sprint, and the product owner and the customers review the deliverable and provide feedback on the build. This feedback could take the form of bugs or changes in functionality which could significantly alter the product backlog and the work done in the next sprint.

Hybrid approaches

Depending on project requirements a specific software development method is selected to fit its

purpose. However, if particular requirements of a project do not fit one specific model, this brings complications for the development method. A common response to these issues is the combination of different plan-driven and agile practices to create a Hybrid development model. Kuhrmann et al. [4] define hybrid approaches as: "*any combination of agile and plan-driven approaches that an organizational unit adopts and customizes to its own context needs (e.g. application domain, culture, process, project, organizational structure, techniques, technologies.*)".

This combination is primarily done to harness the qualities of a model and reduce its weaknesses by incorporating the strengths of another model. A recent study, performed in 2018 by Kuhrmann et al. [4] on Hybrid Software Development Approaches reported that among the 69 surveyed practitioners across Europe, hybrid development methods were widely used, regardless of company size or industry domain. Similarly, Forrester reported the results of their Global Developer Technographics Survey which revealed that 74% of the surveyed teams mix agile to other software development methodologies [7].

In hybrid approaches, *Agility* provides flexibility and space for innovation, *Stability* on the other hand provides high assurance and risk prevention. Hybrid methodologies will be able to increase Customer involvement and meet changing requirements without compromising stability.

Related Work

The idea of hybrid models has already appeared in literature. West [8] proposed the water-scrum-fall approach which suggests that SCRUM is limited to the development-team level, meaning the requirements analysis and release management to follow more traditional approaches. In 2014, Kumar et al. [9] proposed a hybrid approach for requirement engineering in the agile development approach with the help of Joint Application Development (JAD), and the prioritization of the requirements in the agile is helped by the viewpoint. Doshi and Patil [10] proposed a Competitor Driven Development (CDD) model, a hybrid agile process model of Extreme Programming (XP) and Feature Driven Reuse Development (FDRD) agile methods for IT product development. CDD process model is based on self-realizing requirements generation by keeping an eye watch on competitor's upcoming product and market response. The authors suggested Extreme Programming (XP) practices for the development of the identical/better looking product. Tanveer [11] proposed a hybrid approach by combining the best practices of both SCRUM and RUP. In the proposed hybrid approach, nine RUP disciplines have been reduced to 7 disciplines after considering suitable mapping with SCRUM. The product owner creates a product log as a Business requirement model. Sprint plan meeting, daily SCRUM meeting, and sprint review meeting can easily be embedded into elaboration and construction phases. Each sprint boosts with a sprint plan meeting and finalizes with the sprint review meeting. Sprint review meeting can be embedded in the transition phase.

Cooper and Sommer [12] proposed an Agile–Stage Gate hybrid by integrating the elements of both classic stage-gate and agile. The stage-gate process structure and the series of time-boxed sprints of agile development are designed to improve communication and productivity. At the end of each sprint, the project team produces a tangible result that can be demonstrated to stakeholders, including customers, for validation and to identify needed design changes. Sohaib et al. [13] proposed to adapt various design thinking practices into the exploration and planning phases of Extreme Programming (XP). Design thinking practices integrated into the XP exploration phase (Empathy, Define, User persona, Design thinking-user stories); Design thinking practices integration into the XP planning phase (Automated Iterative Prototyping, Prototype Usability Evaluation, User Testing). A hybrid SCRUM Agile Integrated Development framework (SCRUM AID) consisting of the SCRUM and the Integrated Product Development (IPD) approach was proposed by Thiele et al. [14]. This framework takes the five phases of the IPD approach Prototype, Product principle, Pre-Product, Production preparation, and Product release as the superordinate structure to clearly define the intended outcomes of each sprint. The three roles in IPD, marketing, design, and manufacturing are considered together as the SCRUM team and work in parallel on the product. Daily meetings are organized to exchange knowledge and results between the three roles of the SCRUM team. The role of project administration and management in the IPD approach acts as the SCRUM Master.

Hybrid SV-SCRUM Model

SV-Model is an approach that assists developers in producing high-quality software and providing high assurance. Comparatively, SCRUM [6] is a unique agile method focusing on managing iterative development rather than a specific technical approach. SCRUM is extremely beneficial for an organization because it improves existing engineering practices through well-defined workflow, roles, and artifacts [6]. The proposed Hybrid SV-SCRUM Model integrates SCRUM management practices into the SV-Model to expedite product development. This integration is primarily done to harness the qualities of the SV-model and reduce its weaknesses by incorporating the strengths of the SCRUM approach.

The SV-SCRUM Model is divided into two categories, stages that can be executed in a single pass based on the SV-model, and stages that can be executed iteratively based on SCRUM concept of agile development. In the SV-SCRUM Model functional and security requirements move from the SV-Model into SCRUM at its design, implementation, and integration testing, and back into the SV-Model for system security testing and acceptance testing as shown in figure 2.

Since SCRUM is followed to manage the iterative development in the SV-SCRUM Model, this change

of development approach involves several role changes. The mapping of SV-SCRUM model roles with the SCRUM roles is illustrated in table (1). A cross-functional SCRUM team must be carefully formulated to ensure a high level of comfort of interaction between the SCRUM team which can have a positive impact on project development [16]. The level of comfort of interaction between project team members can have a positive or negative impact on the team's morale.

Table 1 Mapping of SV-SCRUM and SCRUM Roles

SV-SCRUM Model Role	SCRUM Role	Description
Business Analyst	Product Owner	Has a clear understanding of customer needs and provide frequent feedback from user perspective during the development process.
Team Leader	SCRUM Master	Maintains the development process, enforce the rules, and keeps the team focused to ensure that the team is functional and productive.
Quality Assurance Manager	Quality Assurance Master	Maintain and manage the quality assurance activities of static and dynamic testing. He ensures the verification and validation takes part in development process to review the quality of the team most important artefacts and deliverables to ensure they meet user and security requirements.
Security Expert	Security Master	Charged in leading security activities. He helps the team in identifying assets, threats and security requirements. This role helps in constructing and executing security test cases.
-Business Analyst -Software Architect -Developer -Tester	SCRUM Team	The development team with skill of self-organizing, self-managing, and cross-functional team. The team member should contribute to all tasks at all stages. The SCRUM team required to work together and pair up with any of team member. The common ways of working together are pair development, pair testing.

Stage 1: Requirements analysis based on SV-Model

The requirement analysis phase of the SV-SCRUM Model is executed in a single pass where the important functional and security requirements are defined upfront. The SCRUM team with the security master and quality assurance master collaborates with the product owner to execute the requirement analysis phase of SV-SCRUM. The main advantage of the diverse requirement engineering team (SCRUM team, security master, quality assurance master, and product owner) is that the team will have multiple viewpoints on the functional and security requirements of the system. Having different eyeballs with a different perspective will lead to better requirements identification and reduce the misunderstanding of requirements and resolve any conflicting issues early before the conflict occurs in later stages of the development process.

In the Hybrid SV-SCRUM Model, the Software Requirement Specifications (SRS) referred to as a product backlog. Quality assurance master and testers with product owner work in parallel with the SCRUM team to design acceptance test cases, functional test cases, security test cases, and risk-based security test cases. When a conflict between the team occurs, the team should engage in collaborative conflict handling for finding a suitable decision through negotiation and compromise facilitated by the SCRUM master.

- **Stage 2: Sprint Planning**

The SCRUM team conducts a sprint planning session and the product owner determines which functional requirements and security requirements in the upcoming sprint. Threat Assessment and Prioritization (DREAD algorithm) assists the security master to choose the highest priority security requirements that are of the highest value to the system to be implemented during the sprint.

- **Stage 3: Development Based on SCRUM**

The SCRUM approach is followed during the design, implementation, and testing of the selected functional and security requirements in the sprint backlog. The usual sprint duration of 4 weeks can be divided into two 2-week mini sprints of development and testing. At the end of 2-week development mini-sprint, the quality assurance master and security master with SCRUM team conduct a mini testing sprint to verify the functional and security requirements. Daily SCRUM meetings at each min-sprint can help in identifying any difficulties to development and testing and resolve them on time.

- **Stage 4: Review based on SV-Model**

At the end of each sprint, the Sprint Review takes place. In the sprint review, the system testing is conducted by the SCRUM team with the security master and the quality assurance master, and the obtained results are compared to the expected results to verify that the functional requirements are met and security requirements mitigate the threats that threaten the system and its assets. The system test scenarios and the results should be demonstrated and explained to the product owner. User acceptance test is performed by the product owner to provide feedback on the deliverable product. If any feedback from the product owner that could take a form of changes in system functionalities the team can revisit the requirements analysis phase. This could significantly alter the product backlog before start planning for the upcoming sprint. This leads to flexibility and ability to accommodate changing requirements. SCRUM team, Security Master, quality assurance master, and the Product Owner perform the Sprint Retrospective Meeting

facilitated by the SCRUM Master to answer mainly two questions: What went well during the last Sprint? What could be improved in the next Sprint? in the just-concluded sprint to determine what could be changed that might make the next sprint more productive.

Comparison of the SV-SCRUM Model with SV-Model and SCRUM

Table 2: Comparison of SV-Model, SCRUM and SV-SCRUM

Properties of model	SV-Model	SCRUM	SV-SCRUM Model
Project Size	Medium to large	Small to medium	Small, medium and large
Focus on teams	No	Yes	Yes
frequent releases of working software	No	Yes	Yes
Support developing security system	Yes	No	Yes
Detailed Documentation	Yes	No	Yes
customer involved in production	No	Yes	Yes
Iterative software development	No	Yes	Yes
Improved visibility	No	Yes	Yes
Engineering practices	Yes	No	Yes
Management and productivity practices	No	Yes	Yes
Accept changes in iteration at any time	No	Yes	Yes

The purpose of table (2) is to show the comparison of SV-SCRUM Model with SV-Model and SCRUM. The comparison is based on the quality parameters adapted from [17, 18] and their level of availability in the models. The list of quality parameters in Table 2 is indicative and not exhaustive.

Conclusion

In this paper, a hybrid secure software development process model named a Hybrid SV-SCRUM Model which integrates SCRUM management practices in SV-Model. This combination is primarily done to harness the qualities of the SV-model and reduce its weaknesses by incorporating the strengths of the SCRUM approach. The main objective of the presented SV-SCRUM model is to develop high-quality secure systems while maintaining iterative development and rapid customer feedback. A comparative analysis between the SV-SCRUM Model, SV-Model, and SCRUM showed that the SV-SCRUM model covers the important quality parameters. Future work is necessary to validate the proposed approach by conducting experiments to evaluate and demonstrate the effectiveness of the proposed SV-SCRUM Model.

References

[1]- Isaias, P., & Issa, T. (2015). Information system development life cycle models. *In High Level Models and Methodologies for Information Systems*, 21-40. Springer, New York, NY.

[2]- Akrout, R. Alata, E. Kaaniche, M., & Nicomette, V. (2014). An automated black box approach for web vulnerability identification and attack scenario generation. *Journal of the Brazilian Computer Society*, 20(1), 1-16.

[3]- Abdulrazeg, A. A., Norwawi, N. M., & Basir, N. (2014). Extending V-model practices to support SRE to build secure web application. *In 2014 International Conference on Advanced Computer Science and Information System*, Kuala Lumpur, Malaysia, 213-218. IEEE.

[4]- Kuhrmann, M., Diebold, P., Munch, J., Tell, P., Trektere, K., McCaffery, F., & Prause, C. R. (2018). Hybrid software development approaches in practice: Europe perspective. *IEEE Software*, 36(4), 20-31.

[5]- Tell, P., Klünder, J., Küpper, S., Raffo, D., MacDonell, S. G., Münch, J., & Kuhrmann, M. (2019). What are hybrid development methods made of? An evidence-based characterization. In 2019 *IEEE/ACM International Conference on Software and System Processes*, Montreal, Canada 105-114. IEEE.

[6]- Hayata, T., & Han, J. (2011). A hybrid model for IT project with Scrum. *In Proceedings of 2011 International Conference on Service Operations, Logistics and Informatics*, Beijing, China, 285-290. IEEE.

[7]- West, D., Grant, T., Gerush, M., & D'Silva, D. (2010). Agile development: Mainstream adoption has changed agility. *Forrester Research Inc*, 2(1), 41.

[8]- West, D. (2011). Water-Scrum-Fall Is The Reality Of Agile For Most Organizations Today. *Forrester Research Inc*.

[9]- Klünder, J., Hebig, R., Tell, P., Kuhrmann, M., Nakatumba-Nabende, J., Heldal, R., & Küpper, S. (2019). Catching up with method and process practice: An industry-informed baseline for researchers. *In 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP),* Montreal, Canada, 255-264. IEEE.

[10]- Doshi, V. P., & Patil, V. (2016). Competitor driven development: Hybrid of extreme programming and feature driven reuse development. *International Conference on Emerging Trends in Engineering, Technology and Science (ICETETS),* 1-6. IEEE.

[11]- Tanveer, M. (2015). Agile for large scale projects: A hybrid approach. *In 2015 National Software Engineering Conference (NSEC),* Rawalpindi, Pakistan, 14-18. IEEE.

[12]- Cooper, R. G., & Sommer, A. F. (2018). Agile–Stage-Gate for Manufacturers: Changing the Way New Products Are Developed Integrating Agile project management methods into a Stage-Gate system offers both opportunities and challenges. *Research Technology Management*, 61(2), 17-26.

[13]- Sohaib, O., Solanki, H., Dhaliwa, N., Hussain, W., & Asif, M. (2019). Integrating

design thinking into extreme programming. *Journal of Ambient Intelligence and Humanized Computing*, 10(6), 2485-2492.

[14]- Thiele, H., Weber, S., Reichwein, J., Bartolo, J. A., Tchana, Y., Jimenez, L., & Borg, J. C. (2020). A scrum agile integrated development framework. *In 2020 proceedings of the design society: design conference,* 747-756. Cambridge University Press.

[15]- Deemer, P., Benefield, G., Larman, C., & Vodde, B. (2007). The scrum primer: An introduction to agile project management with scrum. *Scrum Training Institute*, 1(2).

[16]- Anitha, P. C., Savio, D., & Mani, V. S. (2013). Managing requirements volatility while scrumming within the V-Model. *In 2013 3rd International Workshop on Empirical Requirements Engineering (EmpiRE)*, 17-23. IEEE.

[17]- Anwer, F., Aftab, S., Shah, S. M., & Waheed, U. (2017). Comparative Analysis of Two Popular Agile Process Models: Extreme Programming and Scrum. *International Journal of Computer Science and Telecommunications,* 8(2), 1-7.

[18]- Singh, A., & Kaur, P. J. (2019). Analysis of software development life cycle models. *In Proceeding of the Second International Conference on Microelectronics, Computing & Communication Systems,* 689-699. Springer.

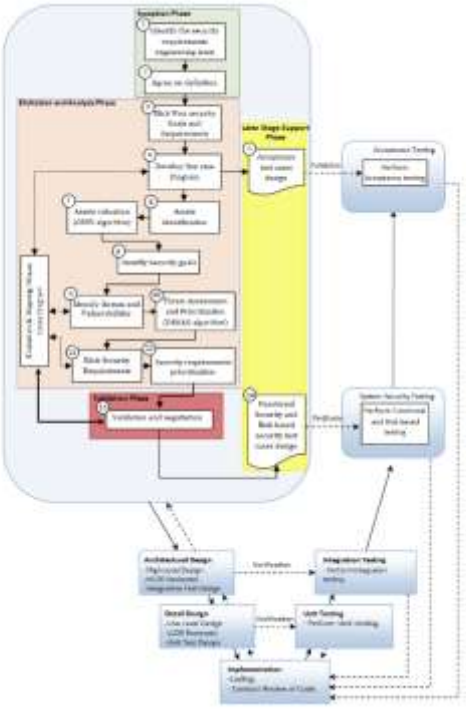

Figure 1 SV-Model

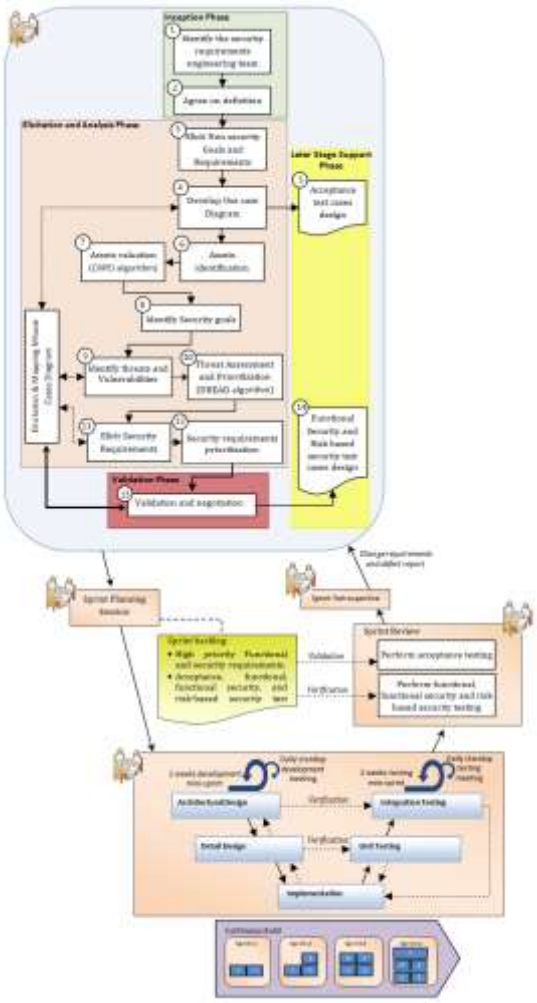

Figure 3 SV-SCRUM Model