Agile Development of Security Critical Software

By Sai Datta Vishnubhotla,
Blekinge Institute of Technology.

ABSTRACT

Today, industries are rapidly adopting agile software development methods, considering their cost-effectiveness and flexibility that facilitates handling new requirements during the contingency of tight deadlines. Security is a critical aspect of software development, especially in the case of financial sector. In traditional software development, security issues were handled by experts. However, in agile development, it is impractical and costly to involve a security expert in every agile team. Therefore, there is a need to extend agile processes with security related quality control and support.

Industries with agile teams, developing security critical software indicate wide differences between teams, in terms of their ability to handle software security risks. Experienced teams with a high level of security competence may be hampered by excessive security related routines, leading to lower software productivity. Whereas in less experienced teams, such routines and procedures are necessary to obtain secure software.

There are various parameters that affect an agile team's ability to handle security related issues in a software, for instance, the educational background, experience of the team members and how the competence of the team members complement each other. However, relevant parameters have to be identified for estimating the security maturity of an agile team.

Our major objective is to define an index that measures a team's security expertise and develop models to estimate the security maturity level of a team, based on team members' experience and training. We consider the case of Ericsson's mobile money system which employs agile methods for overall system development. Further, we focus on building and evaluating a model that predicts how different ways of grouping developers in teams affects the quality and development cost of various program components. The results will be applied and evaluated in different industrial environments.

The developed models assist in estimating how various training and educational activities affects the estimated level of security and the cost for software development. With this knowledge, it is possible to compare training costs with estimated gains in terms of higher security and/or lower development cost.