Evaluation of the HAVOSS software process maturity model

Höst, Martin; Hell, Martin

Published in:
Euromicro Conference on Software Engineering and Advanced Applications (SEAA)

DOI:
10.1109/SEAA51224.2020.00031

2020

Document Version:
Peer reviewed version (aka post-print)

Link to publication

Citation for published version (APA):
Höst, M., & Hell, M. (2020). Evaluation of the HAVOSS software process maturity model. In *Euromicro Conference on Software Engineering and Advanced Applications (SEAA)* (pp. 137-140). IEEE - Institute of Electrical and Electronics Engineers Inc.. https://doi.org/10.1109/SEAA51224.2020.00031

Total number of authors:
2

Accepted for publication in Proceedings 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), 2020.

Evaluation of the HAVOSS software process maturity model

Martin Höst
dept. Computer Science
Lund University
Lund, Sweden
martin.host@cs.lth.se

Martin Hell
dept. Electrical and Information Technology
Lund University
Lund, Sweden
martin.hell@eit.lth.se

***Abstract*—The HAVOSS (Handling Vulnerabilities in OSS) maturity model describes important processes for managing security vulnerabilities in OSS modules in developed products. So far, the model has not been evaluated in any real assessment process. Here we present a study where the model was evaluated by using it in assessments of processes for two product types in one organization. Each assessment was conducted in a focus group meeting where their procedures were analyzed. The evaluation was conducted by posing specific questions about the model during the focus group meetings and by investigating how difficult it was to assess the maturity of practices from the transcribed text. It was found that some practices were easy to assess, while other could be analysed separately for different parts of the products. Further work can be conducted on how assessments can be conducted and how they can be combined with other software security initiatives.**

***Index Terms*—Open Source, Vulnerabilities**

I. Introduction

Many organizations include Open Source Software (OSS) in their code base in order to obtain advantages such as efficient adoption of new technologies and lower development costs. Even if thoroughly tested and widely used and scrutinized code is less likely to contain bugs and security vulnerabilities, basically no code is without bugs and vulnerabilities. The number of disclosed vulnerabilities identified by their CVE (Common Vulnerabilities and Exposures) number is high and increasing. During 2019, the number of new CVEs exceeded 17,000, which is almost a factor three compared to the roughly 6,400 reported in 2016[1]. Thus, in addition to handle security in their own code, organizations also need to handle new vulnerabilities in third party and open source code.

Due to the increased focus on OSS and the increased number of vulnerabilities, the HAVOSS maturity model (Handling Vulnerabilities in OSS) was developed and proposed in [1]. The purpose was to support organizations when they plan how to manage such vulnerabilities in their processes. The model focuses solely on handling vulnerabilities in third party and open source code. This is in contrast to many well known and widely used security maturity models, such as BSIMM [2], SAMM [3] and Microsoft SDL [4]. These models instead focus on the whole organization and the software development lifecycle, including, e.g., in-house developed code and security training. Thus, HAVOSS is a complement to other security maturity models. In this paper, we perform an on-site assessment with the model, evaluating how it can be used in an organization. Through interviews with representatives from the organization, we investigate how well the maturity can be determined for the included areas and practices.

[1]Statistics from https://nvd.nist.gov

The remainder of this paper is organized as follows. In Section II the background in terms of the evaluated model is described and in Section III the research methodology is presented. The results are presented in Section IV together with an analysis, and in Section V conclusions are formulated.

II. Background (The HAVOSS model)

The HAVOSS model is organized as a maturity model, which can be used in assessment of an organization.

A. Capability areas

The model contains six capability areas, where the first can be seen as a pre-condition to the other areas, and the second to the fifth as a series of processes carried out for vulnerabilities, while the last area focuses on internal and external communication activities about vulnerabilities. Below the areas are described shortly. They are described in more detail in [1], where each capability area is described as a set of practices.

1) Product Knowledge (Capability area A): In order to be able to identify vulnerabilities in their developed and maintained products, the company must have information available about which versions of OSS components are included in different versions of their products. It may also be necessary to track which product versions are used in practice.

2) Identification of Sources and Monitoring of Sources (Capability area B): The first step for an organization to take with a vulnerability is to identify it. There are several sources that may be monitored, e.g., the NVD database [5], "blogs", etc. It is necessary to have procedures for identifying sources that are relevant to the organization, and to have procedures for monitoring the identified sources. It may also be necessary to have procedures to receive vulnerability information from external sources that themselves take action to disclose identified vulnerabilities.

TABLE I
MATURITY LEVELS

Level	Description
0	We don't do this.
1	We do this in an ad-hoc way based on individual's own initiatives.
2	We know how we do this, but we do it in different ways in different teams/products.
3	We have defined processes for this that are common to all teams / products.
4	We collect experience and/or metrics from our approach and base improvements on that.

3) Evaluating Vulnerabilities (Capability area C): When vulnerabilities have been identified it is necessary to evaluate their importance for the products that are developed and maintained by the organization, e.g., with respect to severity and probability of being exploited.

4) Remedy of Vulnerabilities (Capability area D): Typically an organization needs separate processes for managing i) vulnerabilities that need urgent remedies, ii) vulnerabilities that need remedies that can be incorporated in the normal update cycles, and iii) vulnerabilities where it is found that no action is needed.

5) Delivering Updates (Capability area E): There is a need for procedures for delivering updates. This can require more effort, e.g., in some IoT systems where it is not possible to distribute updates without manual labour.

6) Communication (Capability area F): Organizations need processes for communication about vulnerabilities, both internally in the organization, and externally, e.g. with customers.

B. Maturity levels

During an assessment, the intention is that every capability area should be assessed as being on a level 0–4 as described in Table I.

C. Prior evaluation of the model

It is inherently difficult to evaluate models like maturity models. For example, it is of course impossible to study two identical organizations and carry out process improvement with both in different ways. However, evaluation of maturity models can be classified in three different ways [6], either i) "off-line" by studying consistency of terms, comparing to process knowledge, etc, ii) by involving practitioners and investigating their views of importance and relevance of factors in the model, or by iii) conducting actual assessments with organizations. In [1], it is reported from evaluations of type i) and type ii) that were conducted while developing the model.

III. RESEARCH METHODOLOGY

The main objective of the research was to understand more about how assessments should be conducted with the model as one step in investigating the suitability of HAVOSS for process improvement with respect to vulnerability management.

The research was conducted in cooperation with an industrial organization that develops embedded systems with hardware and software. The organization has a history of more than 30 years and they have today more than 1000 employees, and they deliver products to customers globally. The majority of the products can be seen as parts of IoT systems, which are developed and maintained by integrators, i.e. not end users or users of consumer products.

The research was carried out by conducting two focus group meetings (e.g. [7]) with the main goal to identify improvement suggestions by doing an assessment with the HAVOSS model, i.e., conducting assessments. That is, the evaluation is most similar to type iii) according to [6]. The two focus group meetings were conducted on different types of products.

- The first meeting focused on their main product, which as described above is an IoT product.
- The second meeting was conducted for a product that can be described as a web-based management system for their products. That is, it consists of web-related technologies, such as JavaScript, and is significantly easier to update than the main product.

The two different types of products make it possible to compare the suitability of HAVOSS for development and maintenance of embedded systems to development and maintenance of web-based services. At each meeting we included about 5 people, with responsibilities of security, of handling vulnerabilities, and in management positions with product responsibilities.

A. The focus group meetings

Each focus group meeting consisted of the following parts:

1) First the basic objectives and the design of HAVOSS were presented to the participants and the agenda of the focus group meeting was presented.
2) For each capability area, each practice was discussed. The goal was to obtain an understanding on how the unit worked with the practice and on what level the unit can be said to be according to Table I.
3) A set of evaluation-questions were asked in order to obtain the unit's view on the questions. The following questions were asked and discussed: "What was relevant and what was not so relevant?" "Is there anything that is unnecessary in the model?", and "What was hard and what was not so hard to answer?"

During the focus group meeting the moderators (i.e., the authors of this paper) showed one question at a time on a "beamer", and directed the conversation so that the current processes and the maturity levels were discussed. During the meetings, the audio was recorded and later transcribed into text format.

The questions in part 2 above were later used to receive a level of practice for the unit. These levels are not presented in this paper.

B. Data analysis procedure

The transcribed text was imported into QSR International's NVivo qualitative data analysis software for coding and analysis. Coding was mainly done by the first author, although some questions were discussed with the second author in this process.

During the analysis we coded the material with codes representing the factors of the model and the maturity level as understood from the text. For the same text chunks as these codes we also coded the difficulty of coding the maturity level, as seen by the researcher. That is, a chunk of text describing a factor in HAVOSS was coded with three codes:

- One for the practice, i.e., one possible code for each question asked by the moderator in step 2 of the meeting.
- One for the maturity level, i.e., one possible code for each maturity level as described in Table I.
- one for the perceived difficulty of coding the maturity level. That is, this describes the subjective opinion of the author for how difficult it was to decide the maturity level of the practice from the coded piece of text. Two different codes were used, 'clear' and 'unclear'.

The motivation for these codes is that we wanted to be able to look at the perceived maturity model from each piece of text, and then look at all text pieces that concerned a certain practice. This makes it possible to assess the maturity of the organization, which however is not the main objective of this paper. We also wanted to be able to analyze how difficult it was to understand the maturity from the texts, as an indication of how easy it is to use the HAVOSS model, i.e., how easy it is to use the model in an assessment.

In addition to the codes above we used the following three codes to evaluate HAVOSS, and the assessment process:

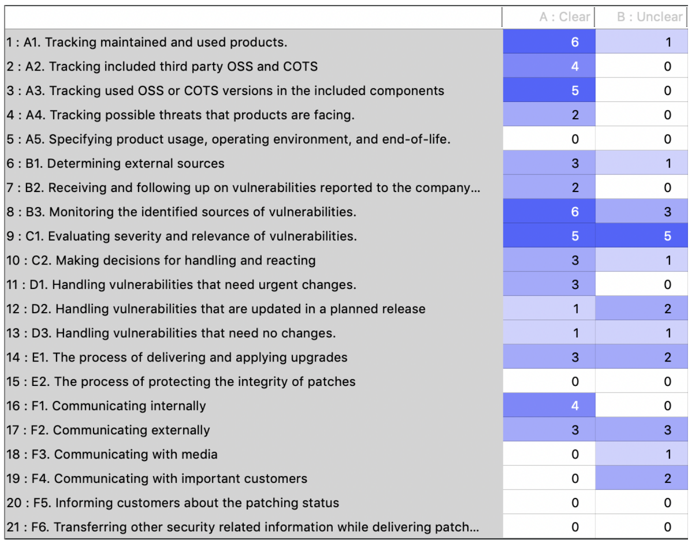
- *Other important factor*, meaning that the discussion at the meeting revolved about something that was important to the participants, but not clearly included in the HAVOSS model.
- *Not important in model*, meaning that the discussion at the meeting showed that the participants did not find the topic of the model important.
- *Process improvement proposal*, meaning that the participants not only discussed the current situation but identified process changes based on the discussion.

The analysis was conducted by investigating both the texts that were coded with practice and maturity, and the codes that were given for texts that concern the evaluation of HAVOSS (other important factors, not important in model, and process improvement proposal).

C. Validity

A main validity threat that has been identified relates to *reliability* (e.g. [8]), i.e., to what extent the results depend on the specific researcher. Both the conduct of the focus group meetings, the coding of the text from the meetings, and the analysis and interpretation may be influenced by the interpretation of the researcher. Care has been taken not to be too subjective or biased in these steps. We have also been two researchers instead of only one, which we believe decreases the risk. However, we believe that further usage of the model, not the least by others, will increase our knowledge of how useful it is and what changes need to be made. This is also related to the *external validity*. The model has only been used in one organization. This is also a reason to use the model in other situations and collect experiences from those occasions.

TABLE II
CROSS TABULATION OF CODING OF PRACTICE VERSUS CODING OF PERCEIVED DIFFICULTY OF CODING MATURITY OF PRACTICE



	A : Clear	B : Unclear
1 : A1. Tracking maintained and used products.	6	1
2 : A2. Tracking included third party OSS and COTS	4	0
3 : A3. Tracking used OSS or COTS versions in the included components	5	0
4 : A4. Tracking possible threats that products are facing.	2	0
5 : A5. Specifying product usage, operating environment, and end-of-life.	0	0
6 : B1. Determining external sources	3	1
7 : B2. Receiving and following up on vulnerabilities reported to the company...	2	0
8 : B3. Monitoring the identified sources of vulnerabilities.	6	3
9 : C1. Evaluating severity and relevance of vulnerabilities.	5	5
10 : C2. Making decisions for handling and reacting	3	1
11 : D1. Handling vulnerabilities that need urgent changes.	3	0
12 : D2. Handling vulnerabilities that are updated in a planned release	1	2
13 : D3. Handling vulnerabilities that need no changes.	1	1
14 : E1. The process of delivering and applying upgrades	3	2
15 : E2. The process of protecting the integrity of patches	0	0
16 : F1. Communicating internally	4	0
17 : F2. Communicating externally	3	3
18 : F3. Communicating with media	0	1
19 : F4. Communicating with important customers	0	2
20 : F5. Informing customers about the patching status	0	0
21 : F6. Transferring other security related information while delivering patch...	0	0

IV. RESULTS

A. Coding of practice versus coding of perceived difficulty of coding maturity

As described above, two focus group meetings were held. The coding of practice and the coding of perceived difficulty are cross tabulated in Table II. The numbers in the table show the number of text chunks in the the material that is coded with each combination. Table II shows the codes from both focus group meetings. The numbers should be used with care and the exact numbers are not as important as the major differences. If, for example, there are very many text chunks that are coded as "unclear" we can take that as an indication of difficulty of coding but one or two instances should not be seen as important.

In Table II it can be seen that it was possible to find text areas in the transcripts for almost all practices. However, there were a few practices for which no text was found. For A5, about specifying product usage, this is an area that was not covered at the meetings. We believe that this has to do with that this is an area that is too far from the responsibilities of the participants. It may be something that is more relevant for more development-related parts of the organization. For the area of E2, about integrity of patches, this too is managed by another part of the organization. Some of the communication-related practices (F5, F6) were not covered at the meeting because of time limitations at the meeting and it did not seem so relevant to the participants at the meeting.

For some practices it was rather easy both to identify the text areas that concerned the practices and to decide which maturity level the practices were at. A1, A2, and A3 deal with rather basic product knowledge, where it was easy for the participants to explain the practices. It was also easy to explain the practice of internal communication. As with A1-A3, we believe that this is seen as a rather basic ability for which they have long experience of.

For some practices there were some text area for which it was seen as easy to identify the maturity level, but there were also text areas for which it was not so easy to identify the maturity. In some cases this had to do with that within the evaluated unit (one on focus group 1 and one in focus group 2) there are different parts of the products that are managed in different ways. As a general example, in a web server, the procedure for managing updates of JavaScript libraries may be different from the procedures for updating the web server as such. At the focus group meetings we saw that, when it comes to identifying sources of vulnerabilities, it was easier for some parts of the developed and maintained product than for other. For some parts of the product there are clear, and accepted, procedures followed by many organizations, while for other parts it requires own defined procedures and more effort.

To summarize, the research indicates that most practices are possible to assess, although they require input from rather many roles in an organization. In this case we involved what we believed to be a broad spectrum of roles, but there were still some areas that were not discussed (e.g. A5 as described above).

Our recommendation is that the assessment procedure should be adapted to this, and maybe different groups discuss different practices. In the same way, it may also be a good idea to present the results of an assessment separately for different parts of the product, which would make it easier to identify process improvement proposals than if only one single result is presented for each practice.

B. Results with respect to usefulness

As described above, we coded text also when we identified discussion pointing out areas seen as important although not included in the model (i.e. code "other important factor"). A recurring discussion in both meetings concerned in-house developed code. Participants thought it was important to study also in-house developed code, not only third party code which is the focus of HAVOSS. As we see it all models cannot include all aspects, and management of third party code is too different from management of in-house developed code to include in-house code in the model. However in an analysis it can be more clearly described that in-house developed code will be handled by other models than what was the case in our focus groups. That is, we see information about this as important, and we suggest that assessments with HAVOSS are combined with other quality improvement work focusing on in-house developed code.

Another need that emerged was to increase ability in security in general and for example to be able to advice customers in general information security aspects like password management and similar. The need is understandable, but we do not propose to include this in a model like HAVOSS.

To summarize, we observed that it was somewhat unclear what the goals of the focus group meetings were compared to other security related improvement work.

Our recommendation is thus that the model should not be used in isolation, but instead be used together with other models in order to capture other security aspects within the organization. The role of this model within this larger context must be clear to the participants. This should include discussing other models, but not necessarily at the same time assessing the maturity according to these.

V. Conclusion

We have done an evaluation of the HAVOSS maturity model by conducting an assessment in two parts of an organization. The interviews included people of different roles. It is clear that it is easy to assess the maturity of product knowledge, but the part of the model covering the actual process from identifying vulnerabilities to delivering updates is more difficult to assess in a consistent way. It is also clear that it is important to include many different organization roles in the evaluation in order to have a wide coverage of the organization. Despite wide coverage, in our assessment, we would have needed even more roles to cover all areas and practices. Finally, we can conclude that it is important to clarify the role of the HAVOSS model in the context of other known maturity models. It should not be used in isolation, but as a complement to more general models with a wider focus on security in the organization. Further research includes more extensive evaluations, e.g. by investigating the quality of products produced by an organization after an assessment and improvement program based on HAVOSS.

Acknowledgment

This research was funded in part by the Swedish Government Agency for Innovation Systems (Vinnova), grant 2018-03965 (project HATCH), and in part by the Swedish Foundation for Strategic Research, grant RIT17-0035.

References

[1] P. N. Bideh, M. Höst, and M. Hell, "HAVOSS: A maturity model for handling vulnerabilities in third party OSS components," in *Product-Focused Software Process Improvement - 19th International Conference, PROFES 2018, Wolfsburg, Germany, November 28-30, 2018, Proceedings*. Springer, 2018, pp. 81–97.

[2] G. McGraw, S. Migues, and J. West, "Software security and the building security in maturity model (BSIMM)," *Journal of Computing Sciences in Colleges*, vol. 30, no. 3, pp. 7–8, 2015.

[3] P. Chandra, "Software assurance maturity model - a guide to building security into software development," OWASP, Tech. Rep., 2017.

[4] Microsoft, "Simplified implementation of the Microsoft SDL," Microsoft Coporation, Tech. Rep., 2010.

[5] NIST, "National vulnerability database," https://nvd.nist.gov/, (visited on: March 01, 2020).

[6] Y. Y. L. Helgesson, M. Höst, and K. Weyns, "A review of methods for evaluation of maturity models for process improvement," *J. Softw. Evol. Process.*, vol. 24, no. 4, pp. 436–454, 2012.

[7] J. Kontio, J. Bragge, and L. Lehtola, "The focus group method as an empirical tool in software engineering," in *Guide to Advanced Empirical Software Engineering*, F. Shull, J. Singer, and D. I. K. Sjøberg, Eds. Springer, 2008, pp. 93–116.

[8] P. Runeson, M. Höst, R. Austen, and B. Regnell, *Case Study Research in Software Engineering – Guidelines and Examples*. Wiley, 2012.