International Journal of Creative Research Thoughts
Volume 1, Issue.4, April 2013
ISSN:-2320-2882

ORIGINAL ARTICLE

SOFTWARE CERTIFICATIONS: A NEW DEMAND OF THE ERA

Sushil Kumar Sahu, Dr.Ashim Ranjan Sarkar
Asst. Professor
Dept. of CS & IT
Christ College, Jagdalpur

ABSTRACT

Software is currently used to control important aspects of our life like medical devices, automobiles, aircraft, manufacturing plants, nuclear generating stations, space exploration systems, elevators, electric motors, automated trains, banking transactions, telecommunications devices and a growing number of devices in industry and in our homes. Software is also used for critical systems. Now we can imagine if the software does not "control" what happens. Obviously, many of these systems have the potential to cause physical harm if they malfunction. Even if they do not cause physical harm, their malfunctions are capable of causing financial and political chaos.

INTRODUCTION:

Certification refers to the confirmation of certain characteristics of an object, person, or organization. This confirmation is often, but not always, provided by some form of external review, education, assessment, or audit. Accreditation is a specific organization's process of certification. Currently there is no consistent regulation of software, and society is starting to demand that software used in critical systems must meet minimum safety, security and reliability standards. Manufacturers of these systems are in the unenviable position of not having any clear guidelines as to what may be regarded as acceptable standards in these situations.

Software Certification: Certification of software is becoming crucial for companies developing mission and safety critical systems. As a result of software-related disasters, some professionals believe that licensing or certification is nowadays inevitable. At present, there is no agreement on what development and assessment methods, techniques, tools, or even evaluation metrics are more suitable to provide evidences on which to base software certification. These process-oriented standards are conceived to suggest strategies and practices to be adopted along the entire development cycle. Certification does not refer to the state of legally being able to practice or work in a profession. This refers to processes intended to determine if a product meets minimum standards, similar to quality assurance. Different certification systems exist in each country. For example, the in Russia it is the GOST R Rostest.

Objectives: Objective of this paper is to believe that standard guidelines should focus not only on process, but also on product properties, and that companies should provide evidences regarding the actual product behavior, rather than their development process. Shifting from process-oriented to product-based certification is enforced by the increasing adoption of third- party (both commercial and open source) software components also in critical system. However, the adoption of off-the-shelf (OTS) software item raises also challenges and difficulties related to their integration, verification, assessment and maintenance, making it very tough to produce evidences that they behave safely. Also this new view is opening totally new challenges with respect to the current reference standards, for both developers and standard organizations.

Challenges of Software Certification - Developing reliable software is a challenging task in itself, but there are several challenges specifically related to certification mentioned as below -

- maintaining high reliability, especially when a combination of diverse development techniques is used,
- minimizing certification efforts, especially for product families and interconnected systems of systems,
- reducing certification and re-certification times,
- linking between artifacts and certificates, and
- providing useful information (e.g., estimates of certification efforts).

Efforts on Software Certification: Researchers and practitioners from both academia and industry, people from government and certification authorities, that work on different aspects of software certification, reports on:

- Methodological improvements of development process of software to be certified.
- Modeling techniques, formalisms, measurements processes, new metrics, techniques, strategies, and tools for:
    - safety analysis, and safety evidences formalization and quantification for certification purposes;
    - design practices conciliating requirements and constraints of certification standards with the needs of modern, complex, component-based systems, as well as with the needs of developer companies;
    - cost-effective Verification & Validation (V&V) specifically oriented towards certification goals;
    - assessment of safety-related qualities in software systems to be certified;

  - safety assessment and evidences formalization, selection, integration, development, and verification of commercial OTS and OSS components in the context of software certification;
  - design, development, V&V, evidences formalization, and assessment to support product-based certification.
- Practical experiences on real case studies regarding certified software or software to be certified.

The Software Certificate: A certificate contains all information necessary for an independent assessment of the properties claimed for an artifact. Obviously, the exact nature of the certificates depends on the nature of the artifact, the property, and the claim. However, a Software Certificate Management System (SCMS) needs a unified view of certificates. At its most abstract, a certificate thus has to represent the three entities involved in the certification process -

- the artifact being certified,
- the property being asserted, and
- the certification authority.

Software Certificate Management System (SCMS): A software certificate management system (SCMS) provides a range of certification services. It maintains the links between different system artifacts (e.g., design documents, engineering data sets, or programs) and different varieties of certificates, checks the validity of certificates, provides access to explicit audit trails, enables browsing of certification histories, and enforces system wide certification and release policies.

Certifiable Artifacts: Certifiable artifacts include not only the conventional software artifacts (e.g., product families, completed systems, individual components, or even code fragments) but also supporting non-software artifacts: requirements documents, system designs, component specifications, test plans, individual test cases, scientific and engineering data sets, and others. In other words, the supporting evidence for one certificate can be considered as the artifact of another certificate. For example, if the correctness of a component is to be certified using traditional black-box testing, the test harness and the test scripts are supporting evidence for the certificate; at the same time, the test harness can itself be certified, e.g., by a code review, and is thus the artifact of another certificate.

Certificate Hierarchies: Certificates for an artifact are not an unstructured collection but exhibit some hierarchical structure. It consist of –

(i) The System Structure: The internal structure of a system is reflected in the certificate hierarchy. If a system is decomposed into a number of subsystems, and each subsystem is built from a number of components, then a certificate for the system depends directly on the

certificates of the subsystems and indirectly on the certificates of all involved components. A system must be able to represent the structure, taking into account language-specific visibility rules like module and subsystem boundaries that can limit the propagation of changes.

(ii) The Certificate Types : The validity of a certificate can also depend on certificates for the supporting evidence or even the authority, e.g., when a code review can only be signed by a certified software engineer. This part of the certificate hierarchy reflects the internal structure and procedures of the organization developing the software.

Certification Services: A SCMS combines the following functionalities of a database and a make-tool. Specifically, it provides a variety of different services as below -

(a) Certificate construction

Given an artifact, a claimed property, and a certification authority, the SCMS attempts to construct the certificate, invoking automatic mechanisms and notifying individuals of pending tasks, as appropriate. It estimates the time and effort that the certification takes.

(b) Editing and revoking

Users can deem an individual certification authority to no longer be valid. The SCMS revokes all certificates which depend on this.

(c) Certificate maintenance

The SCMS carries out intelligent re-certification when a (customizably) appropriate change has taken place in the code or, more generally, software artifacts certifies. Existing (sub-) certificates reuses where possible, especially where product families are concerned.

(d) Auditing Since the SCMS provides a complete certification history with full information about all procedures followed, comprehensive audits can be carried out, applying alternative tools and/or oversight to any elements. The audit itself can then be recorded in the certification database.

(e) Schema management

It must be customizable to existing procedures. It can be thought of as having a client-server architecture. The SCMS is the client and allows users to “plug and play” with arbitrary certificate servers.

Current Technology and Improvement Perceived: A SCMS can build on an existing secure infrastructure. There are a number of differences from existing technology where advances are required:

- linking to software artifacts,
- the wide diversity of forms of certification, both formal and informal, and
- the need for customizability and extensibility.

The SCMS should be an integral part of a development tool suite and use the same underlying data structures. Model-based software development tools should allow the definition of arbitrary domain-specific certificate types with respect to explicit domain models.

Conclusions:

Incremental certification and re-certification of code as it is developed and modified is a prerequisite for applying modern, evolutionary development processes, which are especially relevant for the area where an extremely high accurate data is input for a correct output.

Bibliography –

- Software Certification and Software Certificate Management Systems, http://ti.arc.nasa.gov/m/profile/edenney/papers/Denney-Fischer-softcement05.pdf
- Roger S.Pressman. *Software Engineering A practitioner's Approach*