ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 9, September 2013

CERTIFICATION OF SOFTWARE COMPONENTS FOR REUSE

Neha Malik

Assistant Professor, Dept. of CSE, Dronacharya College of Engineering, Gurgaon-123506, India

ABSTRACT: Certification is the process of verifying a property value associated with something, and providing a certificate to be used as proof of validity. Commercial software could be tagged with certificates that define minimal guarantees about how a software "unit" will behave in the future and under what assumptions it will behave in those manners. A software quality certificate is simply a fact sheet that spells out known software output behaviours and under whatconditions these occur. The objective of this paper is specifying a Software development life cycle with certification.

Keywords: Components, Certification, Framework, Warranties.

I. INTRODUCTION

There are no methods combining the current knowledge of componentselection and component certification as anintegrated part of component-based softwaredevelopment (CBD) process. Several works discussthe lifecycle process [2] divided into (i) componentdevelopment, (ii) component selection and (iii)system development processes inwhich component certification is treated implicitlyor not at all. There are some works that relatecomponent-based development processes to acertification process. In this paper, it isdiscussed whether a certification process is anecessary part of component-baseddevelopment. However, our aim is the necessity ofrecognizing component certification as a separate process, but integrating it into the overall CBD process these four processes inorder to provide a cooperative process that worktogether in order to provide quality aspects aroundthe component's and system's developmentactivities.

II. COMPONENT CERTIFICATION

Component certification is a method to ensurethat software components conform to well-definedstandards; based on this certification, trustedassemblies of components can be constructed. However, this task seems to be very difficultbecause the software engineering community hasexpressed many and often divergent properties to evaluate software components.In addition, existingliterature is not that rich and enough to explain practical aspect of software componentcertification, but some relevant researchexplores the theory of component certification inacademic scenarios. Some terminologies related to Component Certification are:

- **Component Technology**

A component technology consists of a component standard and a language in which to specify component assemblies.

- **Component Framework**

A component framework is a conformant implementation of a component technology.

- **Certification**

Certification is the process of verifying a property value associated with something, and providing a certificate to be used asproof of validity.

III. TRUST AND CERTIFICATION

Trust is a quality attribute we wish to adhere to interactions that transpire in the component marketplace (just as confidentiality is a property we wish to adhere to interactions in our earlier analogy) e.g. when one purchases a light bulb it is expected that the base of the bulb will screw into the socket in such a way that it will produce the expected amount of light. The size and threading has been standardized and a consumer trusts that the manufacturer of any given lightbulb haschecked to make certain that each bulb conforms to that standard within some acceptable tolerance of some set of property values.The interaction between the consumer and the bulb manufacturer involves an implicit trust. In the case of the lightbulb there is little fear that significant damage would result if the bulb did not in fact exhibit the expected property values. This is not the case when purchasing a gas connector. In this case, explosion can result if the connector does not conform to the standard. Gas connectors are certified to meet a standard, and nobody with concern for safety would use a connector that did not have such a certificate attached. Certification is a mechanism by which trust is gained. Associated with certification is a higher requirement for and level of trust than can be assumed when using implicit trust mechanisms.

When we apply these notions to component-based software development, we recognize that it may also be valid to use different mechanisms to achieve trust depending upon the level of trust that is required and the cost associated with providing it.

IV. CHALLENGES

Component selection happens already today in a large scale (evidently, since there is a component marketplace and many systems are built using components). However, component certification is neither completely understood nor established in industrial praxis. The examination of the fundamental differences between component selection and component certification unveil some important challenges to be addressed for softwarecomponent certification to be established.

1. Standardization

Standards are needed so that test and analysis results as well as issued certificates are universally recognized and have an agreed-upon meaning. To this end, and also useful for component selection, analysis and test methods need to be designed enabling a meaningful division and packaging of tests andanalyses between component vendors, certifiers, certification institutes, system developers and software organizations.

2. Costs

It must be considered who will pay for the cost of component certification. For component selection the cost is intrinsic to the software project and must be included in the project plan.

3. Liability & Warranties

Which kind of warranties does the customer have from a component certified in case the component fails during operation? Presumably, if the evaluation context is specified in enough detail it becomes possible to limit the risks for the certifier.

V. CERTIFICATION PROCESS

The certification process is sketched in fig. 1 and explained in the following paragraphs [1]

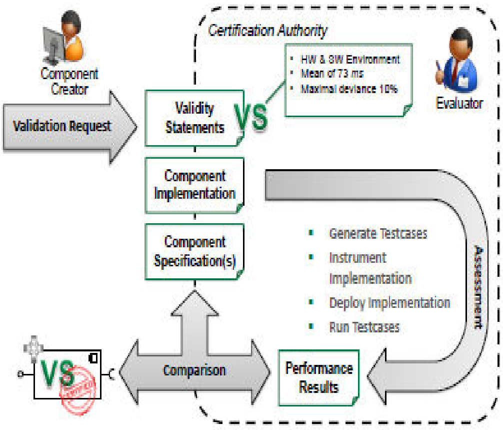

Fig. 1 Overview about Component Certification Process [1]

1. Generate Test cases
In this step, test cases are generated automatically which are used to evaluate the specification. Automatic generation is chosen to ensure reproducibility of validation results and reduce the necessary human effort. If random numbers are used for generating test cases, a pseudo-random number generator can be used to initialize the random numbers. The initialization number has to be stored along the with the validation information. Both directions, specification against executed implementation and vice versa, are checked by the test cases.
2. Instrument Implementation
The implementation itself must be instrumented in order to log the resource demands within the component and correlate them to the resource demand of the specification. Depending on the measurement method either measurement facilities are directly inserted into the code, platform functions of an adapted application or virtual machine container are applied, or operating system functions accessing the scheduled demands are used.
3. Deploy Implementation
The instrumented implementation and mock-ups must then be deployed in the target hardware and software environment. The environment must match the environment specified in the validity statements to allow a meaningful result of the certification. If some properties of an environment are not important for a specification and are hence not stated in the hardware environment, the evaluator can choose an appropriate environment. An example is if a component does not issue requests for a hard disk it is not relevant which hard disk exists in the execution environment.
4. Run Test cases
The test cases derived in the first step are run on the deployed implementation and measurements of the necessary resource demands are gathered. A test case is run until the requested sample size is reached, which can be either specified directly or being calculated by a confidence level. The overhead for storing the measurements themselves should be as small as possible to prevent unwanted side effects on the performance of the implementation. The measurements from all runs of the test cases are denoted as performance results.

VI. SOFTWARE DEVELOPMENT CYCLE WITH CERTIFICATION

After considering the process of certification it can be concluded that discussed process can easily be embedded into traditional cycle of software development [7] as

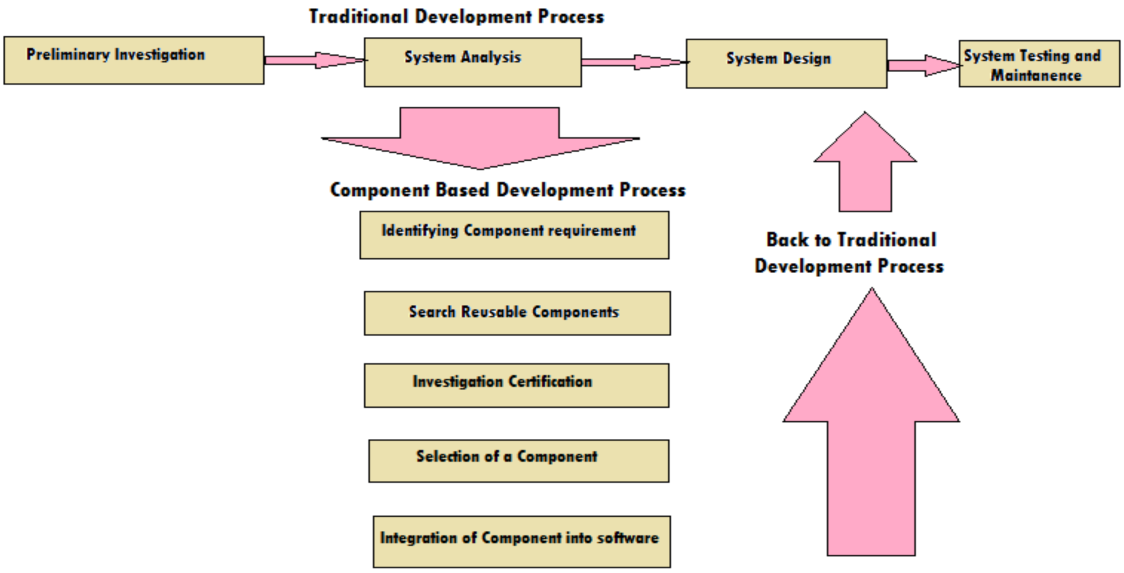
Traditional Development Process
Preliminary Investigation
System Analysis
System Design
System Testing and Maintenence
Component Based Development Process
Identifying Component requirement
Search Reusable Components
Investigation Certification
Selection of a Component
Integration of Component into software
Back to Traditional Development Process

Fig.2 Component based Development process

As shown in figure Software development life cycle(SDLC) that includes certification process also starts as normal process with preliminary investigation, and enter in component based development process in step 4. After that components are identified and searched. After that components are checked for their certification and then selection of component will take place. After integrating the component into software the cycle will move again towards its normal flow.

VI. CONCLUSION

With certification process being included, software developmentstarts with preliminary investigation. Then software moves toward analysis and designing. After that, process will enter in component based development phase. Here identification of component requirements is done. Then searching of reusable components take place. After that developer investigate for adaptations and certification. On the basis of that selection of component will be done. Ultimately component is integrated into the software. And finally the CBSD process will move towards normal cycle of software development.

REFERENCES

[1] Henning, Groenda., “Certification of Software Component Performance Specifications”,FZI ForschungszentrumInformatik,Software Engineering,Karlsruhe, Germany, pp.10-14, 2000.

[2]Stojanovic, Zoran., “An Integrated Component-Oriented Framework forEffective and Flexible Enterprise Distributed Systems DevelopmentSystems”,Engineering Group Faculty of Technology, Policy and ManagementDelft University of Technology Jaffalaan,The Netherlands, pp. 7-12,1999.

[3] Popov, P.,Strigini, L., Riddle, S., and Romanovsky, A.,“Protective Wrapping of COTS Components”, NJIT University, USA.,pp. 168-177, 1997.

[4] Woodman, M.,Benediktsson, O.,Lefever, B., and Stallinger, F., “Issues of CBD Product Quality andProcess Quality”,Annals of Software Engineering, pp.349-414, 1998.

[5] Garlan, D., andSchmerl, B., “Component-BasedSoftware Engineering in Pervasive ComputingEnvironments”, IEEE workshop,pp.403-444, 1996.

[6] Zhong,Sbeng.,“Software Library for Reuse-Oriented Program Development”, University of Windsor, Windsor, Ontario, Canada, pp. 2-20,2000.

[7] Crnkovic, I.,Chaudron, M., and Larsson, S., “Component-Based Development Process and Component Lifecycle”, In International Conference on Software Engineering Advances (ICSEA), pp. 10, 2006.

[8]Voas, J. M., “Certifying Off-the-Shelf Software Components”,IEEE Computer, pp. 9-15,2000.