What Is Software Certification?

Marc Bender

McMaster Centre for Software Certification (McSCert)
Department of Computing and Software
McMaster University

Introduction

- **Motivation**
- **Goals**
- **Approach**

Motivation

- **Vital** question
  - McMaster Centre for □
  - □ Consortium
- **Difficult** question
  - Even the *goal* is not 100% agreed-upon
- **Stimulating** question
  - Leads to productive discussions (or less productive arguments)
  - Attempts to understand and answer it can help organize our ideas and our work

Goals

- **Parse** the question
- **Frame** the answer

Approach

- Separate “what is” into 3 parts:
  - **Definition**
  - **Description**
  - **Vision**
- The rest of the talk has 2 parts:
  1. Present a *decomposition* of the definition
  2. Present a *framework* that encompasses the description and vision aspects

Definition

- **Ontology**
- **Certification**
- **Software**
- **Software Certification**

Ontology

- Arguing over terms and definitions *wastes time*
- We have to use many “loaded words”.
- A formal **ontology** lets us identify and define our *lexicon*.

Obviously, most important terms to define (along with their derivatives):

- Software
- Certification

Certification

- Two aspects
  - The **means**: the certifying *process* or *activity*
  - The **end**: the property or state of something that is *certified*

Important derivatives and variants: certificate, standard, certificate-granting authority, certification process, qualification, licensure.

Software

What is software?

- From an *engineering* point of view
- Ignoring the environment is counterproductive
- Should adopt an *inclusive* definition if possible:
  - Software as *aspect* as opposed to software as *object*.

This might help us get *product-focused*.

Important derivatives to define: product, engineering, artifact, software-intensive system, documentation, verification, validation, environment.

Software Certification

- As the means:
  - Certification of systems containing software *scoped to the software aspects*.
- As an end:
  - Certification of systems comprising entirely software?
  - Recommendation: don’t use software certification in this sense.

Framework

- **Evidence**
- **Confidence**
- **Criteria**
- **Determination**
- **Certification**

Evidence

- The facts, objects, documents, measurements
- Possible categorization:
  - Direct (material) evidence.

Test results, proofs of correctness, etc.

  - Indirect (circumstantial) evidence.

Adherence to process, personnel qualification, etc.

- **Empirical**

Confidence

- The *interpretation* of evidence.
- The (formal or informal) *corroboration* of criteria.
- The “progress bar” of certification in practice.
- **Epistemic**

Criteria

- The *requirements* of the evidence (and confidence).
- Roughly, the “checklist” for certification.
- The “musts”, “oughts”, “shoulds”, and “mays”.
- **Deontic**

Determination

- The practical/actual process of certification
  - Who does what?
  - How is the evidence actually evaluated?
  - How is the certification process documented, monitored?
  - What happens in the case of *rejection*?
  - How is the *certificate* assigned, etc.?
- **Pragmatic**

Certification

- Two decompositions:
  - Standard vs. certificate
    - Standard defines what happens for assignment of *any* certificate.
    - Certificate is assigned to one specific artifact.
  - Vindicative vs. indicative
    - Vindicative: the *internal* view, representing the sum of the evidence, confidence, criteria and determination leading to the certification.
    - Indicative: the *external* view: what the certification means to the outside world, including qualitative aspects.
- Certification is a signifier or *symbol*.
- **Semiotic**

Conclusion

- **How Does This Help?**
- **Future Work**

How Does This Help?

- Helps us make our suggestions more concrete:
- “Be more product-focused” becomes
  - $\rightarrow$ Consider direct evidence,
  - $\rightarrow$ Weigh direct evidence more strongly (in confidence),
  - $\rightarrow$ Establish/incorporate criteria about direct evidence.
- Helps understand and organize the task at hand: designing a certification scheme everyone can agree on.

Future Work

- Develop an initial generic model of certification that “instantiates” the generic framework
- Delve more deeply into each of the five aspects of certification
- Apply the framework to existing models of certification as a method of critique

End