Abstract—Software plays a key role in high-risk systems, i.e., safety and security-critical systems. Several certification standards and guidelines, e.g., in the defense, transportation (aviation, automotive, rail), and healthcare domains, now recommend and/or mandate the development of assurance cases for software-intensive systems. As such, there is a need to understand and evaluate (a) the application of assurance cases to software, and (b) the relationship between the development and assessment of assurance cases, and software engineering concepts, processes and techniques. The ICSE 2013 Workshop on Assurance Cases for Software-intensive Systems (ASSURE) aims to provide an international forum for high-quality contributions (research, practice, and position papers) on the application of assurance case principles and techniques for software assurance, and on the treatment of assurance cases as artifacts to which the full range of software engineering techniques can be applied.

Index Terms—Software engineering, assurance cases, safety, security, certification, argumentation, evidence.

I. INTRODUCTION

An assurance case is a “reasoned and compelling argument, supported by a body of evidence, that a system, service or organization will operate as intended for a defined application in a defined environment” [1]. It is typically developed at the system level and focuses on a particular property, e.g., safety, security, or more generally, dependability.

Increasingly, software plays a key role in critical, high-risk systems. The current practice to demonstrate that software does not adversely affect system properties, such as safety, is to satisfy a set of objectives prescribed by standards and/or guidelines. A good example of this is the DO-178C guideline recommended for software assurance in airborne systems. Typically, such documents recommend and/or mandate the processes to be employed for assurance, and the artifacts to be produced as evidence. However, the rationale connecting the recommended assurance processes, the artifacts produced, and the system properties to be assured, is largely implicit.

Assurance cases provide exactly the mechanism, i.e., arguments and argument structures, to make these connections explicit and, thereby, subject to assessment.

Several standards in high-risk domains such as defense [2], aviation [3], rail [4] and automotive [5] now require the development of an assurance case for software-intensive systems. As such, there is an urgent need to develop models, techniques and tools that target the development of assurance arguments for software. Ultimately, these assurance arguments will form a core part of the assurance case for the wider system.

Assurance cases are typically generated in a phased manner and relate to the main activities in a software engineering process (requirements development, design, implementation, verification and deployment). There are noteworthy parallels between the design of software as an artifact, and the design of an assurance argument for software, as a corresponding artifact:

• For instance, although an assurance argument is not executable, we design an assurance argument to satisfy certain criteria, e.g., maintainability, in a manner similar to the design of software to satisfy similar, corresponding criteria.

• We can modularize and/or abstract arguments [6] for supporting compositional certification, in the same way as modularization and abstraction is applied in software for compositional purposes.

• We can refactor argument structures to improve argument comprehension and maintainability, similar to software code and model refactoring.

• Argument structures exhibit cross-cutting concerns, patterns [7], and properties and constraints, similar to software.

• Formal techniques can be brought to bear on the design, analysis, and creation of arguments [8], [9].

II. WORKSHOP THEME AND GOALS

The theme of the workshop will be the relation between assurance cases and software. Not only do we want to apply assurance case principles and techniques to software assurance, but also we want to treat assurance cases as artifacts to which the full range of software engineering techniques can be applied.

The main goals of the workshop are to:

(1) explore techniques for the creation and assessment of assurance cases for software-intensive systems,

(2) leverage, adapt and apply techniques, concepts, and tools from software engineering in the assurance case lifecycle,
(3) identify the dimensions of effective practice in the development and evaluation of assurance cases, and
(4) identify critical research challenges and define a roadmap for future developments.

III. PAPERS

Eight papers have been accepted, covering three different themes that address the workshop goals:

A. Principles

- C. Weinstock, J. Goodenough, and A. Klein “Measuring Assurance Case Confidence using Baconian Probabilities”.

B. Notations and Techniques

- P. Conmy and K. Attwood. “Nuanced Term-Matching to Assist in Compositional Safety Assurance”.

C. Applications

- A. Ray and R. Cleaveland. “Constructing Safety Assurance Cases for Medical Devices”.

IV. COMMITTEES

A. Organizing Committee

Ewen Denney, SGT / NASA Ames Research Center, USA
Ibrahim Habli, University of York, UK
Tim Kelly, University of York, UK
John Knight, University of Virginia, USA
Ganesh Pai, SGT / NASA Ames Research Center, USA

B. Program Committee

Robin Bloomfield, City University, UK
Luke Emmet, Adelard, UK
Richard Hawkins, University of York, UK
Kelly Hayhurst, NASA Langley Research Center, USA
Michael Holloway, NASA Langley Research Center, USA
Daniel Jackson, MIT, USA
Insup Lee, University of Pennsylvania, USA
Peter Lindsay, University of Queensland, Australia
Tom Maibaum, McMaster University, Canada
Nikolai Mansourov, KDM Analytics, USA
Robert Martin, MITRE Corporation, USA
Yutaka Matsuno, Nagoya University, Japan
Roger Rivett, Jaguar Land Rover, UK
Christel Seguin, ONERA, France
Kenji Taguchi, AIST, Japan
Fredrik Törner, Volvo Car / Chalmers University, Sweden
Alan Wassung, McMaster University, Canada
Robert Weaver, Airservices Australia, Australia
Charles Weinstock, SEI, USA

C. Additional Sub-Reviewers

Anaheed Ayoub, University of Pennsylvania, USA
Andrew King, University of Pennsylvania, USA
Mark Nicholson, University of York, UK
Andrew Rae, University of York, UK
Shaohui Wang, University of Pennsylvania, USA
Toshinori Takai, AIST, Japan

V. CONCLUSION

We would like to thank all authors for their submissions, and congratulate those authors whose papers were selected for inclusion into the workshop proceedings. We also acknowledge and thank the program committee members, and their additional sub-reviewers for the time and effort they spent towards preparing an exciting program for this first workshop.

VI. REFERENCES