y is neither feasible nor reasonable since

Workshop Theme
The disciplines of requirements engineering (RE) and software architecture (SA) are fundamental to the success of software projects. Even through RE and SA are often considered separately, the Twin Peaks model acknowledges that treating RE and SA separately is neither feasible nor reasonable since requirements and architecture impact each other. Requirements are constrained by what is technically and economically feasible. On the other hand, feedback from the architecture requires renegotiating architecturally significant requirements with stakeholders. The topic of bridging RE and SA has been discussed in both the RE and SA communities, but mostly independently. Therefore, the motivation for this ICSE workshop is to bring both communities together to explore the state of the art in research and practice, methods, technologies and tools, and to identify emerging trends related to the transition and the relationship between RE and SA.

Paper categories
Position papers (2-4 pages): Discuss on-going research, new challenges and emerging trends, novel solutions and inspiring, providing an divisionally ideas, or propose potential directions for future research and future trends.
Full papers (6-8 pages): Report innovative and original research.
Industrial papers (up to 6 pages): Describe industrial experience, case studies, challenges, problems and solutions.
Experience reports (4-6 pages): Present case studies, empirical experiences with a particular RE or SA practice, pattern or technique.
Education and training papers (4-6 pages): Describe experiences, approaches and tools for teaching RE and SA topics (including lesson plans, assignments).

Main Topics
TwinPeaks 2014 will focus (but not limit itself) on two main topic categories:
1. Techniques and Practices for Twin Peaks: Capturing and maintaining relationships between requirements and architecture; ensuring consistency and traceability between requirements to architecture; creating and evolving trace links; reverse engineering to reconstruct design decisions, architecturally significant requirements and business goals; managing concerns and requirements and architecture viewpoints; support software engineering decision making.
2. Twin Peaks in Context: Twin Peaks appears in different contexts and across different development and project environments. For example, agile development requires lightweight techniques for integrating architectural thinking into the agile process. Another example is software product line engineering which imposes challenges on ensuring consistency across domain requirements, product line architectures, product requirements, product architectures and evolving product lines. Other possible contexts include for example model-driven engineering, safety-critical systems, embedded systems development.

Workshop Goals
TwinPeaks 2014 will provide a venue for researchers, practitioners and educators from the areas of requirements engineering (RE) and software architecture (SA) to discuss their experiences, forge new collaborations, and explore innovative solutions that address the challenges intertwining RE and SA.

Important dates:
Submission date: January 14, 2014
Notification: February 17, 2014
Camera-ready: March 14, 2014
Workshop: June 2014