Requirements Engineering at Age 20:
Looking Back, Looking Ahead

Panel Chair:
Barry Boehm, USC

Panelists:
Mack Alford, Alford Enterprises
Eliot Chikofsky, DMR Group
Tom De Marco, Atlantic Systems Guild
Dick Fairley, SEMA
Matthias Jarke, TH Aachen

The panelists will look back at the progress in requirements engineering since its emergence as a full-fledged sub-discipline of software engineering in 1976. This was primarily marked by a set of special sessions on software requirements engineering at ICSE 2 in October, 1976, followed by the publication of some of the key ICSE 2 papers in the January 1977 IEEE Transactions on Software Engineering. The panelists will then look forward to address the likely evolution of requirements engineering over the next 20 years.

The panelists are exceptionally well qualified to do this. Most were playing key roles in the emergence of requirements engineering in 1976. All are leaders in addressing how requirements engineering should address the emerging world of COTS, cyberspace, objects, applets, and alternative process models.

Below is some background on early developments in requirements engineering, and some candidate issues for the panelists to address during the session.

Looking Back

Clearly, there were significant contributions to software requirements engineering before 1976. The classic 1956 and 1961 papers on software development by Benington and Hosier both emphasized the importance of good requirements definition, and included software requirements reviews as key life-cycle milestones. Requirements definition and traceability tools were emerging in the early 1970's.

However, these were not generally recognized as being in the mainstream of software engineering. The influential 1968-69 NATO software engineering workshops generally partitioned the field into Design, Code, and Test. They stimulated some emphasis on software specifications, particularly formal approaches such as the Vienna Definition Language, but did not draw any significant distinctions between requirements and design specifications.

The main contribution of the 1976 papers was to provide evidence that requirements engineering techniques were applicable to large software systems, and that they paid off. The Teichroew et al. PSL/PSA system, the Alford et al. SREM system, and the Ross et al. SADT system had all been applied successfully to large complex systems. And there was quantitative evidence that they caught significant software problems that would have been much more costly to fix later.

Looking Forward

Recent developments in software development technology and practice indicate the need to rethink various aspects of requirements engineering. The emergence of large commercial-off-the-shelf (COTS) capabilities is creating a 180 shift in the traditional model of software developments: from a requirements-determine-capabilities model to a capabilities-determine-requirements model. Graphic user interface (GUI) builders and applications generators create large software domains in which an application can be developed in less time than it takes to specify the requirements. Domain engineering and product line families of programs create new dimensions of challenge for requirements engineering techniques.

Some candidate issues for the panelists to address are:

- How will requirements engineering techniques adapt to the trends above?
- What will be the roles of formal and informal requirements specifications?
- Will we ever master requirements for quality attributes (reliability, portability, performance, etc.) and their tradeoffs?
- Will we ever master characterization of the software system’s environment?
- How will people factors best be captured in requirements engineering?