Chapter III

# Software and Systems Engineering: Conflict and Consensus

Rick Gibson
American University, USA

## ABSTRACT

*This chapter will identify the key aspects of software engineering and systems engineering in an effort to highlight areas of consensus and conflict to support current efforts by practitioners and academics in both disciplines in redefining their professions and bodies of knowledge.*

*By using the Software Engineering Institute's Capability Maturity Model – Integrated (CMMI$^{SM}$) project, which combines best practices from the systems and software engineering disciplines, it can be shown that significant point of agreement and consensus are evident. Nevertheless, valid objections to such integration remain as areas of conflict. This chapter will provide an opportunity for these two communities to resolve unnecessary differences in terminology and methodologies that are reflected in their different perspectives and entrenched in their organizational cultures.*

## INTRODUCTION

With software an increasingly significant component of most products, it is vital that teams of software and systems engineers collaborate effectively to build cost-effective and reliable products. This chapter will identify the key aspects of software engineering and systems engineering in an effort to highlight areas of consensus and conflict to support current efforts by practitioners and academics in the both disciplines in redefining their professions and bodies of knowledge.

## BACKGROUND

In response to increasing concerns about software development failures, the Software Engineering Institute (SEI) pioneered a software process improvement model in 1988, with the fully developed version of the Capability Maturity Model for Software (SW- CMMâ) appearing in 1993. The key processes were identified as (Paulk, Curtis, Chrissis & Weber, 1993): Requirements Management; Integrated Software Management; Software Project Planning; Software Product Engineering; Software Project Tracking and Oversight; Intergroup Coordination; Software Subcontract Management; Peer Reviews; Software Quality Assurance; Quantitative Process Management; Software Configuration Management; Software Quality Management; Organization Process Focus; Defect Prevention; Organization Process Definition; Technology Change Management; and Training Program.

Since the early 1990s, there have been comparable improvement models introduced in the system engineering community as well, some of which have been published and widely accepted: Systems Engineering Capability Maturity Model (SE-CMM) also known as the Electronic Industries Alliance Interim Standard (EIA/IS) 731, Systems Engineering Capability Model (SECM) and the Integrated Product Development Capability Maturity Model (IPD-CMM). In 1995, SEI recognized 18 process areas in systems engineering. The processes fell into three categories: engineering process, project process and organizational process (Bates Kuhn, Wells, Armitage & Clark, 1995). The resulting avalanche of models and standards has been described by Sarah Sheard (Software Productivity Consortium) as a "Framework Quagmire." In December of 2000, the SEI initiated CMMI$^{SM}$ project, which combines best practices from the systems and software engineering disciplines. (Note: CMM® and CMMI$^{SM}$ are copyrights and service marks of the Software Engineering Institute.)

## ISSUES AND CONTROVERSIES

There is great hope that the SEI initiative will provide the impetus to overcome some long-standing discipline boundaries. The nature of the systems and software