



SOFTWARE ENGINEERING

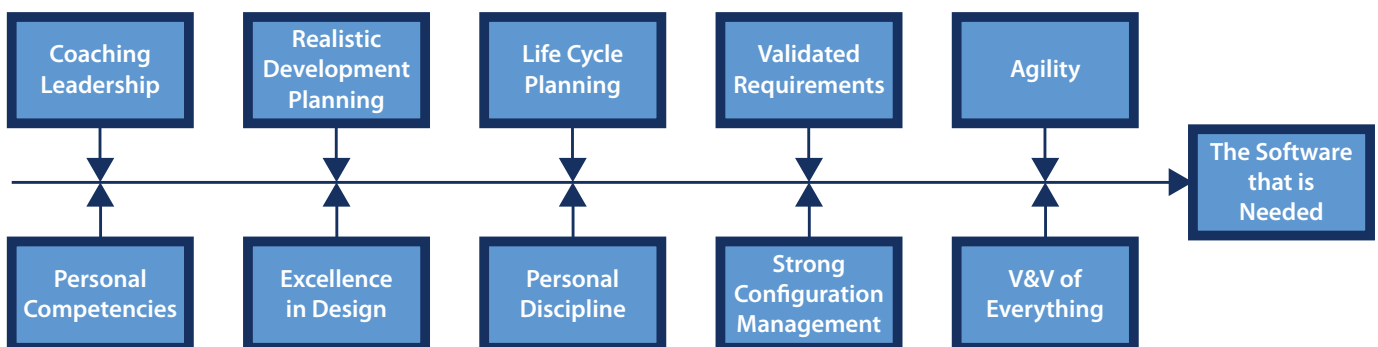
THE DIFFERENCE BETWEEN THE 16% OF LARGE SOFTWARE PROJECTS THAT SUCCEED AND THE...

5-DAY COURSE

... 84% THAT FAIL OR ARE SUCCESS-CHALLENGED IS ENGINEERING (STANDISH GROUP, 2018). LEARN HOW TO MAKE THE DIFFERENCE.

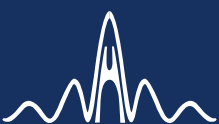
As anybody who has developed software knows, developing the right software correctly is challenging. The high failure/cancellation rate of software projects is a testament. This course approaches software development from a proven engineering perspective: the application of sound principles and efficient methods, having their foundations in life-cycle-based systems thinking. Applicable to the engineering of software both small and large, simple and complex, all training subject areas provide actionable advice and methods for reliably creating cost-effective software solutions.

Additional practices that should be overlaid in the engineering of critical software (safety, security, or mission) are also addressed in this leading course. The course will assist anybody who defines, develops, maintains or manages software or software-intensive systems to better perform their roles.



"The personal experience of the presenter helped in understanding the course material as he was able to put things into perspective/practice. It has backed up a lot of my views and gave me confidence in my working role."

- delegate, EADS, United Kingdom



1. Introduction and Overview

- introduce the presenter and go over the learning methodology
- an overview of course structure and methodology
- general introduction into software engineering
- history of software development, recent trends, the current state and beyond the current state
- an engineering approach to software development including concurrent engineering, systems methodology and thinking
- key role-players in a typical software development environment
- life cycle characteristics and typical stages
- process fundamentals and development models
- sequential versus incremental and iterative development models
- different software development methodologies and their applicability
- the performance of different software development methodologies
- lean software development and value-driven design
- tailoring and process improvement principles

2. Technical Processes

- requirement analysis
 - system requirements, system boundaries, hierarchy and subsystems
 - requirement quality attributes and other fundamentals

- requirement documentation, natural language, UML, and storyboards
- requirement parsing
- concept of use documents
- software tools
- common pitfalls in performing RA
- software design
 - design fundamentals
 - architectural styles and patterns
 - evolutionary architecture
 - design methods
 - software design notations and presentations
 - documentation and tools
- software construction
 - dealing with complexity
 - standards for coding
 - process
 - assessing quality
- system and software integration
 - integration strategies
 - system oriented architecture
 - web services
 - communication protocols
 - interface controls
 - pitfalls and pointers
- validation and verification
 - fundamentals
 - formal, informal, technical, design, code, requirement and other reviews
 - testing
 - other V&V methods
- transition (transfer between different owners)
- operations, maintenance and support
- disposal or retirement

3. Project Processes

- project management: frame works, planning, assessment and control
- estimation and costing
- risk management
- configuration management
- change management
- quality management
- release and deployment
- information management
- human elements and building effective teams

4. Agreement Processes

- overview and contract models
- acquisition
- supply

5. Enterprise-Level Processes

- project portfolio management and program management
- life cycle model management
- quality management
- human resource management
- knowledge management
- enterprise tools

6. Specialty Fields

- This section briefly looks at specialty fields, providing enough information to appreciate the significance and be able to interact with experts in the field. Topics include:
 - engineering of trusted/high integrity systems
 - software life cycle cost analysis
 - interoperability
 - usability analysis and human system integration

To register please visit our website or call our friendly registration team:



PO Box 2385
Ringwood North
Victoria, 3134 Australia



+61 3 9876 7345



contact@ppi-int.com



www.ppi-int.com



PROJECT PERFORMANCE
INTERNATIONAL