

SOFTWARE ENGINEERING

Five Day Course

This course will help you build software-intensive systems that satisfy the need of your enterprise to delight its customers, while developing software at a predictable cost in a predictable time frame. We cover all the essential processes that must be mastered to succeed in this aim. Our goal is to improve your personal, team and organizational performance by applying proven effective principles, development frameworks and methodologies.

Who Should Participate in This Course?

This course is for people who are responsible for specifying, acquiring, developing, evaluating, supporting and/or managing software-intensive systems, for example:

- Software Development Team Leaders
- Software Developers
- Software Engineers
- Programmers
- System Engineers
- System Engineering Managers
- Project Managers of Software-Intensive Projects
- Verification & Validation Managers
- Software Configuration Managers
- System Safety Managers
- Software Maintainers
- Software Procurement Managers

This course is certified by ECSA for 5 CPD points (ref. INCOSE 13/003/15)



This course is recognized by Engineers Australia for CPD purposes (40 Hours)

Attendance at public courses and on-site delivery in Australia may be eligible for SADI funding.



Head Office, Australia
Tel: +61 3 9876 7345
Fax: +61 3 9876 2664
PO Box 2385, Ringwood North
Vic, Australia 3134

United States of America
Tel: +1 888 772 5174
Fax: +1 888 772 5191

Brazil
Tel: +55 11 3958 8064
Fax: +55 12 3212 5582

United Kingdom
Tel: +44 20 3608 6754



contact@ppi-int.com

For more information visit our website: www.ppi-int.com

COURSE OUTLINE

0. Introduction – Why Software Engineering?

- software intensive project issues and challenges
- relationship between problem definition and stakeholder satisfaction
- the case for applying a systematic approach to software systems development
- **workshop - review of software intensive projects and their outcomes**
- principles for software systems development

1. Systems Engineering Life Cycle Processes

- definitions of system engineering
- systems concepts
- systems engineering lifecycle processes
- waterfall, incremental and iterative lifecycle models
- Agile methods
- relationship to software engineering lifecycle processes
- standards and guidelines - ISO 9001, IEEE 1220, EIA/IS-632, EIA 632, ISO/IEC 15288, SEBOK
- organizations – IEC, IEEE, INCOSE, ISO, SEI

2. Software Engineering Life Cycle Processes

- definitions of software engineering
- software engineering concepts
- relationship to systems engineering lifecycle processes
- organizations – IEEE, ISO, SEI, PMI®
- standards and guidelines - ISO 12207, IEEE 12207, IEEE 1012, ISO 90003, SWEBOK, ISO 15504 (SPICE), ISO 15939, ISO 25000 series, CMMI for Development

3. Software Systems Engineering Process Frameworks

- the 12207 process framework
- the CMMI® process framework
- the SWEBOK knowledge areas

4. Software Development Methods and Techniques - Agile, TSP, RUP, etc.

- development approaches explained – Team Software Process (TSP), Extreme Programming (XP), Test Driven Development, Feature Driven Development, Dynamic Systems Development Method (DSDM), Rational Unified Process
- performance comparison of 13 software development methodologies – development cost, development time, software quality, software cost of ownership over 5 years
- the objectives of Agile methods
- the Agile Manifesto
- Agile ten years on
- Agile project management
- Agile success factors, challenges and benefits

5. Project Management Frameworks

- PMBOK®
- PRINCE2TM
- the ISO/IEC 12207 project management processes
- the CMMI® project management knowledge area
- The SWEBOK software engineering management knowledge area

6. Quality Management Frameworks

- ISO 9001
- ISO 9000-3
- the ISO/IEC 12207 QA processes
- the CMMI® QA knowledge area
- the SWEBOK quality management knowledge area

7. Requirements Analysis

- software requirements fundamentals
- requirements elicitation
- requirements analysis
- requirements specification
- requirements validation
- practical considerations
- requirements analysis techniques
- requirements documentation
- requirements management support tools
- **workshop - requirements analysis**

8. Software Design

- software design fundamentals
- key issues in software design
- architectural models
- architectural views and frameworks
- commonly used architectures
- design methods
- software design notations and representations
- design documentation
- design support tools
- **workshop – documenting designs**

9. Software Construction

- conquering code complexity with structure
- ISO 9001 requirements for coding
- coding process
- assessing code quality
- developing coding standards

10. System Integration

- integration
- Service Oriented Architecture (SOA)
- Web services
- SOAP and XML
- interface control
- pitfalls and pointers in system integration

11. Software Maintenance

- software maintenance fundamentals
- key issues in software maintenance
- maintenance techniques
- software maintenance support tools

12. Verification and Validation

12.1. Technical reviews

- software walkthroughs and inspections – Fagan, Gilb
- formal and informal reviews
- requirements reviews
- design reviews
- other reviews
- audits
- administration of technical reviews
- technical reviews and incremental builds
- customer involvement in technical reviews
- pitfalls in conducting technical reviews
- **workshop – reviewing requirements**

12.2 Testing

- software testing fundamentals
- test planning
- test Levels
- test techniques
- test related measures
- test documentation
- software testing tools

12.3 Other Verification and Validation Methods

- control flow analysis
- demonstration
- prototyping
- simulation

12.4 Independent Verification and Validation

- what is IV&V?
- why and when do you use IV&V?
- IV&V requirements and activities
- IV&V agents
- implementing IV&V

13. Software Engineering Management

13.1 Project Management Activities

- project initiation
- scope definition
- project planning
- project monitoring, controlling, and reporting
- project closure

13.2 Software Development Plans

- Software Development Management Plans and Project Management Plans
- relationships of an SDMP/PMP to other plans
- content of the PMP
- content of the SDMP
- pitfalls in preparing a SDMP/PMP

13.3 Scope Management

- defining a project's scope
- types of WBS
- why the WBS is a foundation of effective engineering management
- guidelines for preparing a WBS
- WBS decomposition and work packages
- relationship of a WBS to cost accounts
- WBS development pitfalls and pointers
- **workshop - developing a WBS**

13.4 Time Management

- schedule drivers
- activity definition
- activity sequencing
- activity duration estimation
- schedule development
- schedule control
- **workshop – developing a schedule**

13.5 Cost Management

- cost drivers
- cost estimation techniques
- cost estimation models
- Earned Value Management
- **workshop - developing a cost estimate**

13.6 Risk Management

- the nature of risk
- components of risk
- key risk management activities
- **workshop - developing a risk assessment**

13.7 Configuration Management

- configuration management fundamentals
- configuration identification
- configuration control
- configuration status accounting
- configuration auditing
- configuration management standards
- configuration management pitfalls and pointers

13.8 Release and Deployment Management

- release management
- deployment planning
- Business Continuity planning
- Post Implementation Reviews

13.9 Quality Management

- the origins of quality management
- software quality fundamentals
- software quality characteristics
- quality management characteristics
- quality management tools
- practical considerations

13.10 Software Performance Measurement/Metrics

- technical performance measures
- the Practical Software Measurement (PSM) approach
- the ISO 15939 Software Measurement Process standard
- the SEI Goal Driven Measurement approach
- Function Point Analysis

14. Process Evaluation and Improvement

- process definition
- process implementation and change
- ISO 15504 (SPICE)
- process assessment
- process improvement

15. CMMI®

- CMMI® terms and concepts
- improvement and CMMI®
- CMMI® model representations – staged, continuous
- CMMI® disciplines and process areas
- appraisals and benchmarking
- tailoring

16. Software Acquisition

- ISO 12207 acquisition process
- the CMMI® - Acquisition

17. Engineering of Trusted Software Systems

- recognizing and classifying safety risks
- the risk reduction process
- safety-related system architectures
- quantifying safety integrity
- software safety lifecycle activities

18. Effective Team Management

- motivation
- emotional intelligence
- teamwork
- leading a development team
- coaching a development team

19. Course Review

- Lifecycle models and process frameworks
- Software Engineering technical activities
- Software Engineering management activities
- Process evaluation and improvement
- Trusted software systems
- Teams
- Course closure

helping projects succeed...