



International Institute for Software Process

Attention:

# Software Process Improvement Professionals

Achieve the Prestigious Designation of:

## The International Software Process Improvement Certification (ISPIC)



The only education-based certification for International Software Process Improvement Professionals



International Institute for Software Process

For additional information visit:  
[www.SPInstitute.org](http://www.SPInstitute.org)

## International Software Process Improvement Certification (ISPIC)

### Rationale

Software process improvement has become an essential activity for software organizations in order to meet the challenge of complex software systems and the increasing demand for more reliable systems. Given the broad range of approaches, methods and tools for Software Process Improvement (SPI), organizations are struggling to qualify individuals who can participate in the overwhelming task of improving processes around the organization. There are several certification programs that certify individuals in very specialized areas of software process improvement. Examples are Six Sigma, CMM/CMMI Lead Assessor certifications, and the SEI's SEPM Certificate Programs.

To that end, the International Institute for Software Process (IISP) Advisory Board, a group of industry experts in software process improvement, has recognized the need for a comprehensive education-based certification program that is based on a well-rounded Body of Knowledge. The Educational program supporting the International Software Process Improvement Certification (ISPIC) credential will provide those who participate in SPI efforts with the knowledge and skills necessary to perform their job duties with the highest levels of professionalism and competency. This certification will also help hiring managers qualify individuals who are supposed to participate in any SPI effort.

### The Formal Education Requirement

**Course of Study:** To be awarded the ISPIC credential, a candidate must complete a total ten days of training. The course of study must contain at least one day of training in each of the seven Core Knowledge Areas (CKAs) of the SPI Body of Knowledge (SPIBOK) and days of training in any three Application Knowledge Areas (AKAs) of the SPIBOK as fits within the candidate's job responsibilities and interests.

The Formal Education Requirement must be completed in no more than three years. Credit for any training will expire three years after its completion.

**ISPIC Training:** Candidates may select courses offered by IISP either in a classroom setting, online courses, or any of the one-day tutorials offered in conjunction with ICSPIC conferences. Candidates are required to complete a written exam for each course and pass with a level of performance no less than 80%.

**Transfer Credit:** An ISPIC candidate may transfer work performed outside the ISPIC program for credit towards the ISPIC certification under any of the following conditions:

1. An ISPIC candidate who has been awarded the *Project Management Professional (PMP)* certification, *Certified Software Test Professional (CTM)* certification, or the *ASQ Certified Software Quality Engineer (CSQE)* certification may receive a credit for TWO days towards the CKAs of the SPIBOK.
2. An ISPIC candidate may receive credit for courses conducted by providers other than IISP for a maximum of TWO days towards the AKAs of the SPIBOK. Candidates must submit evidence of

successful completion from a recognized training institution, along with detailed course material for evaluation and a certificate of completion.

### **The Job Experience Requirement**

In order for the ISPIC certification to be granted, a candidate must have a total of at least three years working in software projects, including at least one year in a SPI related role. This requirement must be completed by the time ISPIC is granted. This requirement shall be met by means of a letter of support describing the candidate's specific role and responsibilities over a period of three years or more. The letter must be authored and signed by:

1. The candidate's current or former supervisor/manager
2. The candidate's client or customer (if self-employed)

Multiple sources may be submitted to cover the three-year period. Any variation from this requirement must be reviewed and approved by the IISP Chairperson.

### **Re-certification and Continuous Education**

A person who achieves the ISPIC Certification will need to renew the certification every three years. The requirements and criteria for renewing the certification will be discussed by the Advisory Board at a later date.

## **Software Process Improvement Body of Knowledge (SPIBOK)**

### **Purpose and Intent**

The purpose of the SPIBOK is to define knowledge areas that must be mastered by any Software Process Improvement (SPI) Professional in order to be effective in identifying, documenting and improving software processes. The SPIBOK consists of *Core Knowledge Areas* (CKAs) and *Application Knowledge Areas* (AKAs). All SPI Professionals need to be familiar with and understand key aspects of all CKAs. AKAs represent specific aspects of the software process improvement effort, which an SPI professional needs to know more about, based on his/her experience and the types of organizations and projects with which he/she works.

The intent is for the SPIBOK to be used in two ways:

1. As a guide for organizations and individuals to identify and assess the knowledge and skills needed by SPI professionals to perform effectively.
2. As the basis for the formal education requirements for the education *International Software Process Improvement Certification (ISPIC)*.

### **Core Knowledge Areas (CKAs)**

An effective SPI professional needs to demonstrate familiarity with and understanding of key aspects of the following CKAs.

## **1. Defining and Documenting Software Processes**

- a. Concepts and definition of “Process” and “Software Process”
- b. Relation of process to predictability, improvement, and control over one’s results
- c. Issues involved with software process definition and improvement
- d. Generic methods for identifying and documenting procedural aspects of software processes
- e. Characteristics, strengths, and weaknesses of prominent formal methods for identifying and documenting procedural aspects of software processes (e.g. ETVX, IDEF-0)
- f. Issues and methods for identifying and documenting non-procedural aspects of processes

## **2. Measuring Software Processes**

- a. General concepts and issues of software measurements
- b. Core project, product, and process measures
- c. Measuring people-oriented aspects of software processes
- d. Methods and issues involved in collecting, organizing, and accessing measurements
- e. Identifying an appropriate set of metrics to support organization and project needs (e.g. GQM, PSM)
- f. Analyzing measurements to identify process improvement opportunities (e.g. Trend Analysis, Defect Causal Analysis)

## **3. Evaluating Software Process Capability and Effectiveness**

- a. Process capability measurement concepts and techniques
- b. Directly evaluating software processes and effectiveness
- c. Characteristics, strengths and weaknesses of prominent formalized indirect methods, such as model-fitting or for evaluating software processes, including SCAMPI & CMMI®, ISO 9000 & TickIT, Six Sigma
- d. Auditing and sampling process performance to determine compliance of activities with documented procedures

## **4. Planning and Managing an SPI Program and SPI Projects**

- a. Roles, responsibilities authority, accountability, and skills for managing the overall SPI Program
- b. Methods for establishing and maintaining senior management sponsorship for the SPI Program and SPI projects
- c. Justifying and initiating SPI projects
- d. Tasks, skills, and techniques for planning, organizing, directing, and controlling a software process improvement project (including Risk Identification and Assessment)
- e. Evaluating the effectiveness of SPI projects
- f. Dealing with Cultural, Organizational, and People Issues concerning an SPI Program and SPI projects (including identifying opinion leaders, key people in the SPI process, building consensus, dealing with subtle or overt resistance to change, etc.)

## **5. Implementing Software Process Change**

- a. Organizational change management
- b. Pilot testing and rolling out process changes in organizations
- c. Training people to adopt new processes
- d. Communicating and gaining support for changes involved with the software process improvement process as well as for specific software process improvements and their associated improvement projects

- e. Institutionalizing continuous software process improvement attitudes and behaviors in the organization
- f. Evaluating effectiveness of software process change
- g. Dealing with cultural, Organizational and People Issues concerning software process change.

## **6. Survey of Software Engineering Topics**

- a. Software Requirements Definition and Management
- b. Software Development, Integration, Release, and Support
- c. Software Configuration Management
- d. Software Project Management and Planning
- e. Software Risk Management

## **7. Survey of Quality Assurance and Testing Topics**

- a. Software Quality Assurance and Testing, defined and distinguished
- b. Establishing and operating a Software Quality Assurance Function
- c. Inspections and other forms of static reviews
- d. Requirement Reviews
- e. Levels and types of testing
- f. Dynamic testing techniques
- g. Test Process concepts and issues
- h. Test Management concepts and issues
- i. Defect Tracking and Reporting

## **Application Knowledge Areas (AKAs)**

The following AKAs represent examples of additional topics which are components of some software processes with which an SPI professional may need to familiarize, depending on organization and project specifics. This list of AKAs is not intended to be exhaustive. It is expected to expand as additional relevant topics are identified.

### **1. Requirement Engineering and Management**

- a. Defining and validating requirements
- b. Managing and maintaining requirements
- c. Building a Requirement-Management Process

### **2. Software Development, Integration, Release, and Support**

- a. Traditional SDLC
- b. Agile development
- c. Incremental delivery
- d. Methods for determining and documenting the architecture and design (high-level and detailed) for software products
- e. Methods and activities associated with the release and installation of software systems
- f. Activities associated with supporting and maintaining software products
- g. Activities involved in choosing, contracting with, and assuring the performance of software suppliers (both those who provide commercial off-the-shelf COTS products, and those that do custom development)

### **3. Configuration Management**

- a. Principles and activities of the Discipline of Software Configuration Management
- b. Identifying configuration items
- c. Establishing baselines
- d. Controlling change
- e. Establishing and maintaining repositories
- f. Assuring the integrity of software work products.

#### **4. Project Management and Planning**

- a. All activities associated with planning projects including size, effort and cost estimation, schedule development, resource planning, knowledge and skills planning, etc.
- b. All activities associated with managing projects including comparing actual measures to estimates, determining project status, managing risks, reporting status, taking corrective action, tracking action items

#### **5. Risk Management**

- a. Risk Analysis Methodologies
- b. Risk Identification and Classification
- c. Risk Prioritization and Ranking
- d. Calculating costs and probability
- e. Risk Reporting
- f. Monitoring and controlling risks
- g. Contingency Planning and Mitigation

#### **6. Verification and Validation Methods**

- a. Unit, integration, System, and User Acceptance Level Testing including planning and design activities of each
- b. Inspections and other forms of peer reviews
- c. Code Analysis
- d. Independent V&V

#### **7. Agile Software Development Methods**

- a. The values, principles and philosophies that underpin Agility
- b. Contrasting Agile philosophy to other iterative and incremental lifecycles
- c. Contrasting Agile with the waterfall model
- d. Agile methods and method tailoring
- e. Agile methods and project management
- f. Suitability of Agile methods
- g. Progressive requirements elaboration
- h. Iterative planning and adaptation
- i. Incremental product delivery
- j. Coaching self-directed teams
- k. Agile project monitoring
- l. Welcoming project change
- m. Interpreting Agile practices relative to reference models (e.g. CMMI or PMBOK)
- n. Criticism

#### **8. Software Quality Assurance**

- a. All activities related to establishing an SQA group
- b. Stakeholder Identification

- c. Developing SQA plan
- d. Establish Budget
- e. Establish Personnel
- f. Establish Mission/Objectives
- g. Selling SQA to Management
- h. Demonstrating ROI for SQA efforts
- i. Base lining current software quality levels and Cost of Quality
- j. Defining standards, procedures, methodologies, best practices and guidelines
- k. Instituting metrics and measurements
- l. Deploying processes
- m. Evaluating Methodologies and Automated Tools
- n. Implementing Defect Studies

### **9. Balanced Scorecard**

- a. Basic concepts of the balanced scorecard. How it can be used to improve organization performance
- b. How the balanced scorecard applies to different types of organizations
- c. How to develop more meaningful performance measures
- d. The correct sequence of steps that are necessary to build a strategy-focused organization
- e. How a scorecard system can drive a performance-based budget and employee accountability.
- f. The Learning & Growth Perspective
- g. The Business Process Perspective
- h. The Customer Perspective
- i. The Financial Perspective
- j. Measurement-Based Management
- k. Double-Loop Feedback
- l. Outcome Metrics
- m. Management by Fact

### **10. Software Measurement & Analysis**

- a. Developing a metrics framework
- b. Choosing the best metrics for your situation
- c. Goal/question /metric paradigm
- d. Goal Selecting appropriate measures
- e. Organizational measures/goals
- f. Development organization measures/goals
- g. Quality organization measures/goals
- h. Implementing measurement programs
- i. Comparing your performance to industry standards
- j. Automating metrics collection, analysis and reporting
- k. Integrating measurement and analysis into planning and management reporting
- l. Measurements Objectives
- m. Measurement Specifications
- n. Data Collection and Storage
- o. Measurement Analysis
- p. Measurement Communication

*More AKAs may be approved by the Advisory Board as appropriate.*

# Software Process Improvement Certification



International Institute for Software Process

Attention:

## Software Process Improvement Professionals

Achieve the Prestigious Designation of:

## The International Software Process Improvement Certification (ISPIC)

For additional information visit:

[www.SPInstitute.org](http://www.SPInstitute.org)



International Institute  
for Software Process

636 Mendelssohn Avenue North  
Golden Valley, MN 55427