

Introduction to Software Architecture, its Principles and Methods

Duration

2 days

Instructor

Philippe Kruchten

Class Limit

20 students

Prerequisite

None

Price

On-site

Please contact SPC
for pricing (contact
information on page 2)

Public Training

\$995 (2 days)

*Discount available for
early registration

Materials Provided

- Student manual containing the course slides
- Student handouts with class exercises

The architectural design of a system serves as a bridge between requirements and implementation. Software architecture is the development product that has the potential to give the highest return on investment in terms of quality, schedule and cost. Because its architecture appears early in a product's lifetime, getting it right sets the stage for everything to come – development, integration, testing, and modification.

Introduction to Software Architecture, its Principles and Methods examines the role of architectural design in the system development lifecycle and reviews underlying principles, concepts and methodologies. Questions discussed during this two-day seminar include:

What is software architecture?

Software architecture is intuitively easy to understand, but very hard to define. This course will look at the intricate relationship between software architecture and software design, software development, and system development.

What is the value of software architecture?

What are the economic, business drivers behind software architecture? What are the benefits of having a focus on software architecture? What good does it do to a project? Which projects will benefit most?

How should software architecture be represented?

One approach to better define software architecture is to define a way to represent it, to document it, to make it an objective, tangible artifact that we can transmit, examine, discuss. But architecture of a software intensive system is a rather complex, and most architectural representation use multiple coordinated views. Can UML be used to represent architecture? How, and what elements of UML? What is the use of an architecture description language (ADL)?

Who are the architects?

Who plays the role of the software architect? How does this role differ from software developer or project manager? What are the skills and competences of a software architect? How should software architects be trained?

How is an architecture created or selected?

Where does architecture fit in the development process and lifecycle? Is there an architectural process - a systematic way of developing, validating, assessing an architecture? How does software architecture fit in an iterative development lifecycle? A number of architectural design approaches will be reviewed.

How is an architecture validated?

How is an architectural review and assessment performed? How should an architectural prototype be tested?

TRAINING

Introduction to Software Architecture, its Principles and Methods

Instructor

Philippe Kruchten is a professional software engineer with 30 years of industrial experience developing large-scale software-intensive systems in the areas of telecommunications, aerospace, defense, transportation and software tools. He is responsible for the development of the software development method, the Rational Unified Process®. In addition to RUP, Philippe has also developed a model for the representation of software architecture, based on multiple coordinated views, which led to an IEEE standard.

Philippe has participated in the design of two telephone switches, command and control systems, and most notably led the design of the Canadian Automated Air Traffic Control system. An accomplished author, his most recent and notable publications include a book on the RUP (170,000 copies in 10 languages, so far), an undergraduate textbook on software engineering, developed jointly with a Canadian university professor, and a textbook on RUP with an IBM colleague. Philippe is currently a professor of software engineering at the University of British Columbia.

Intended Audience

This course is ideally suited to software architects, engineers, developers, system analysts, project and product managers and other IT professionals interested in learning more about software architecture and design.

Learning Topics

- Architectural representation
- Desirable qualities of a software architecture
- The role of architecture in managing risk
- Common architectural frameworks
- Object-oriented approaches
- The impact of commercial off-the-shelf products (COTS) and standards
- How and where architectural design fits in an iterative software development lifecycle, such as the Rational Unified Process (RUP)

Outline

Day 1

- Definitions of software architecture
- Benefits of software architecture
- Software architecture in the development lifecycle

Day 2

- Software architecture representations, multiple views, use of UML
- Software architecture artifacts
- A typical architectural iteration

For more information on this or other SPC Springboard courses, please visit www.spcspringboard.com or e-mail SPC at info@spc.ca

Software Productivity Center Inc.
Suite 460 - 1122 Mainland Street
Vancouver, BC V6B 5L1

Vancouver: 604.662.8181 Toll Free: 1.877.548.1948

Fax: 604.689.0141

