



DIGILEAF INC.

Leading Excellence Among Fellows

Course Outline

Enterprise Architecture Learning Track

Software Architecture – A Developer’s Perspective

Designing software given a vague set of requirements and a blank sheet of paper is a good skill to have, although not many people get to do this on a daily basis. With agile methods encouraging collective ownership of the code, it's really important that everybody on the team understands the big picture. To do this, you need to understand why you've arrived at the design that you have. In a nutshell, everybody on the team needs to be a software architect.

This is a two-day training module about pragmatic software architecture. This module will present what "just enough" up front design is, how it can be applied to software projects and how to communicate the big picture through a collection of simple effective sketches. Aimed at software developers, it fills the gap between software development and high-level architecture that probably seems a little "enterprisey" for most developers.

Benefits

This module is useful if any of the following scenarios sound familiar:

- I'm not sure what software architecture is about and how it's any different from design.
- I don't understand why we need "software architecture."
- My manager has told me that I'm the software architect on our new project, but I'm not sure what that actually means.
- I want to get involved in designing software but I'm not sure what I should learn.
- I've been given some requirements and asked to design some software, but I'm not sure where to start.
- I need to make some major enhancements to my system, but I'm not sure where to start.
- I've been asked to write a software architecture document but I'm not sure what to include in it.
- I'm not sure who to talk to in my organization about how best to integrate what we're building.
- I understand what software architecture is all about, but I'm not sure how to tackle it on my project.
- My project seems like a chaotic mess; everybody is doing their own thing and there's no shared vision. Help!

Duration 2 day(s)



DIGILEAF INC.

Leading Excellence Among Fellows

Course Outline

Enterprise Architecture Learning Track

Software Architecture – A Developer’s Perspective

Topics

Day 1

- I. Introductions and Expectation Setting
- II. Software Architecture Defined
 - a) Architecture and Software Architecture
 - b) Design Versus Architecture
- III. The Software Architecture Function
 - a) Software Architects in the Development Team
 - b) Software Architecture Role Versus Lead Developer Role
 - c) Other Roles and Responsibilities
 - i. Coding Architects
 - ii. Soft Skills Needed by Software Architects
 - iii. Avoiding “Ivory Towers”, Collaboration and Coaching
 - iv. Quality Attributes
 - v. Constraints
 - vi. Principles
 - vii. Designing Software Given a Blank Sheet of Paper
- IV. Software Design Exercise (1st Iteration)

Day 2

- I. Visualizing Software
 - a) Reviewing the Diagrams from the Software Design Exercise
 - b) UML vs. Informal Sketches
 - c) Ineffective Sketches
 - d) A Lightweight Approach for Effective No-UML Sketches
- II. Software Design Exercise (2nd Iteration)
- III. Documenting Software Architecture
 - a) Importance of Documentation
 - b) Project and Documentation Stakeholders
 - c) A Lightweight Approach to Software Documentation
 - d) Alternative Approaches for Documenting Software
- IV. Software Architecture in the Development Life Cycle
 - a) Software Architecture in Waterfall, Iterative, and Agile Approaches
 - b) “Just Enough” Up Front Design
 - c) Risk-storming
 - d) Evaluating Software Architecture and Prototypes
 - e) Guidance for Doing “Just Enough” in the Real World