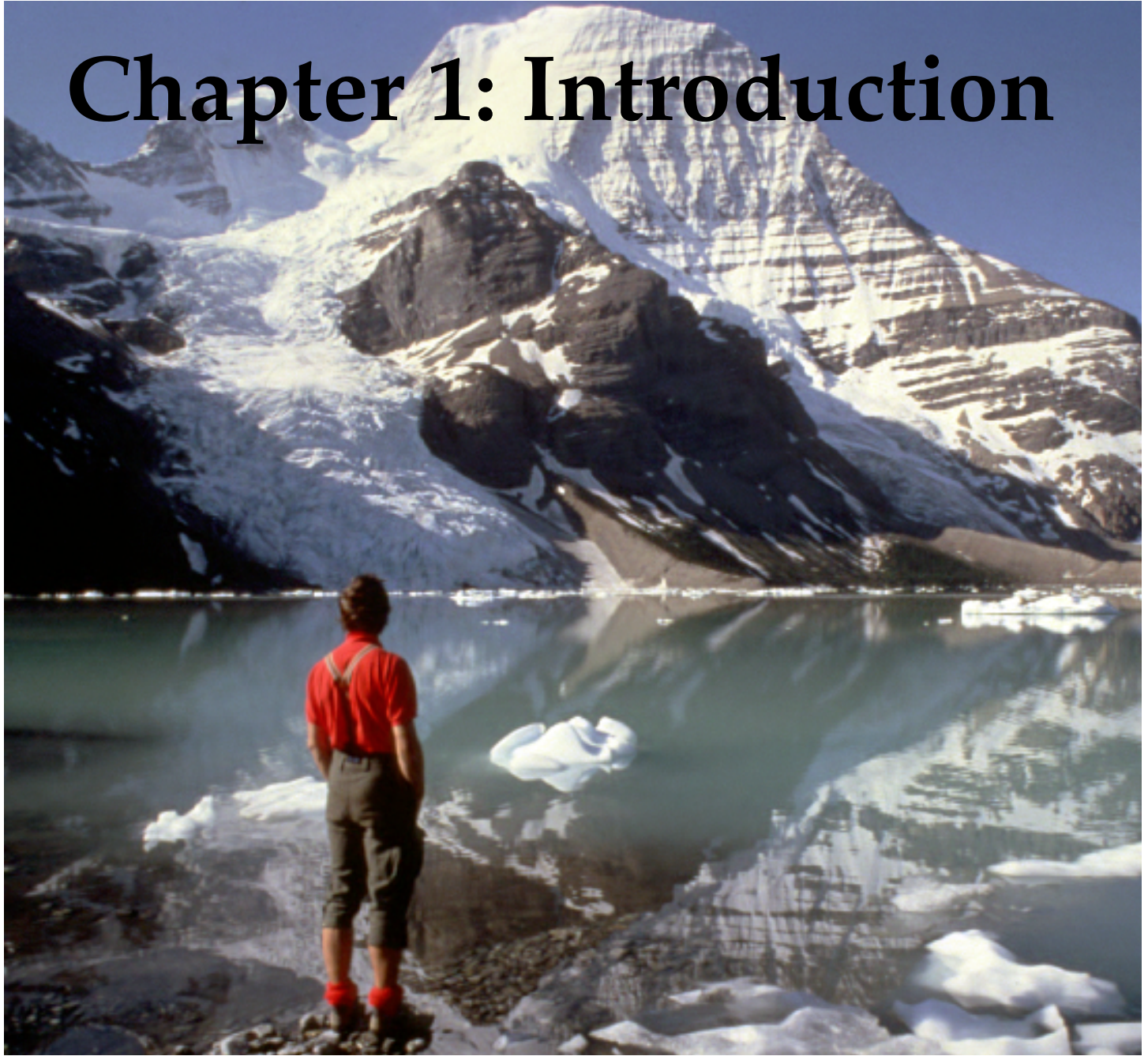


# Chapter 1: Introduction



# Objectives of the Lectures

- Appreciate the Fundamentals of Software Engineering:
  - Methodologies
  - Process models
  - Description and modeling techniques
  - System analysis - Requirements engineering
  - System design
  - Implementation: Principles of system development

# Assumptions for this Class

- Assumption:
  - You have taken an undergraduate Software Engineering course
- Beneficial:
  - You have had practical experience with a large software system
  - You have already participated in a large software project
  - You have experienced major problems.

# Focus: Acquire Technical Knowledge

- Different methodologies (“philosophies”) to model and develop software systems
- Different modeling notations
- Different modeling methods
- Different software lifecycle models (empirical control models, defined control models)
- Different testing techniques (eg. vertical testing, horizontal testing)
- Rationale Management
- Release and Configuration Management

# Acquire Managerial Knowledge

- Learn the basics of software project management
- Understand how to manage with a software lifecycle
- Be able to capture software development knowledge (Rationale Management)
- Manage change: Configuration Management
- Learn the basic methodologies
  - Traditional software development
  - Agile methods.

# Why is software development difficult?

- The problem domain (also called application domain) is difficult
- The solution domain is difficult
- The development process is difficult to manage
- Software offers extreme flexibility
- Software is a discrete system
  - Continuous systems have no hidden surprises
  - Discrete systems can have hidden surprises!

# Software Engineering is more than writing Code

- Problem solving
  - Creating a solution
  - Engineering a system based on the solution
- Modeling
- Knowledge acquisition
- Rationale management

# Techniques, Methodologies and Tools

- **Techniques:**

- Formal procedures for producing results using some well-defined notation

- **Methodologies:**

- Collection of techniques applied across software development and unified by a philosophical approach

- **Tools:**

- Instruments or automated systems to accomplish a technique
- CASE = Computer Aided Software Engineering



# Computer Science vs. Engineering

- **Computer Scientist**
  - Assumes techniques and tools have to be developed.
  - Proves theorems about algorithms, designs languages, defines knowledge representation schemes
  - Has infinite time...
- **Engineer**
  - Develops a solution for a problem formulated by a client
  - Uses computers & languages, techniques and tools
- **Software Engineer**
  - Works in multiple application domains
  - Has only 3 months...
  - ...while changes occurs in the problem formulation (requirements) and also in the available technology.

# Software Engineering: A Working Definition

Software Engineering is a collection of techniques, methodologies and tools that help with the production of

*A high quality software* system developed with a given *budget* before a given *deadline* while *change* occurs

Challenge: Dealing with complexity and change

# Course Themes

## Dealing with Complexity

- Notations (UML, OCL)
- Requirements Engineering, Analysis and Design
  - OOSE, SA/SD, scenario-based design, formal specifications
- Testing
  - Vertical and horizontal testing

## Dealing with Change

- Rationale Management
  - Knowledge Management
- Release Management
  - Big Bang vs Continuous Integration
- Software Life Cycle
  - Linear models
  - Iterative models
  - Activity-vs Entity-based views

# Software Engineering: A Problem Solving Activity

- **Analysis:**

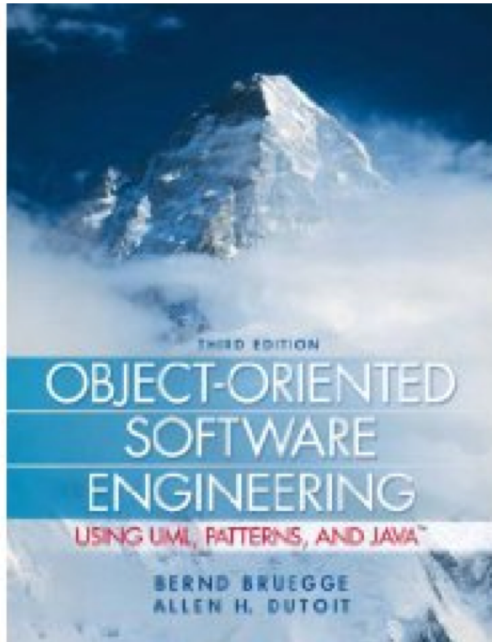
- Understand the nature of the problem and break the problem into pieces

- **Synthesis:**

- Put the pieces together into a large structure

For problem solving we use techniques, methodologies and tools.

# Textbook



*Bernd Bruegge, Allen H. Dutoit*

Object-Oriented Software Engineering:  
Using UML, Patterns and Java, 3<sup>rd</sup>  
Edition

Publisher: **Prentice Hall**, Upper Saddle  
River, NJ, 2009;

ISBN-10: 0136061257

ISBN-13: 978-0136061250