The Inception Phase

• Major goal – develop enough understanding of requirements to decide whether it’s worthwhile to continue
  – Feasibility
  – Buy or build?
  – High-level vision of project
  – Rough estimate of resources required (people, time, money)
  – Major risks
Definition of Risk

• “a variable that, within its normal distribution, can take a value that endangers or eliminates the success of a project” (Kruchten)
• “an unwanted event that has negative consequences” (Pfleeger)
• “something you’d prefer not to have happen” (Somerville)
Types of Risk

- **Project risks** affect project schedule or resources
- **Product risks** affect the quality or performance of the software being developed
- **Business risks** affect the organization developing or procuring the software
Risk Evaluation

• **Risk control** is the degree to which we can change the outcome.
  – Direct and indirect risks

• **Risk impact** is the loss associated with a risk

• **Risk probability** measures the likelihood of a risk being realized

• **Risk exposure** =
  \[ \text{RiskImpact} \times \text{RiskProbability} \]
Risk Management

• Risk assessment
  – Risk identification
  – Risk analysis
  – Risk prioritization

• Risk control
  – Risk reduction
  – Risk management planning
  – Risk resolution
Boehm’s Top Ten Risk Items

1. Personnel shortfalls
2. Unrealistic schedules or budgets
3. Developing the wrong software functions
4. Gold plating
5. Developing the wrong user interface
6. Continuing stream of requirements changes
Top Ten Risk Items (cont’d)

7. Shortfalls in externally performed tasks
8. Shortfalls in externally furnished components
9. Real-time performance shortfalls
10. Straining computer science capabilities
Requirements: Introduction

• A requirement is a capability which the system *must* have or a condition to which it *must* conform.

• The Unified Process focuses on *managing* requirements.
  – Changes in requirements – UP expects and *embraces* change as a fundamental driver of project
  – Finding (eliciting) requirements
Types of Requirements

FURPS+ model

- Functional requirements
- Usability requirements
- Reliability requirements
- Performance requirements
- Supportability requirements
Other Types of Requirements

The + in FURPS+

- Implementation requirements
- Interface requirements
- Operations requirements
- Packaging requirements
- Legal requirements

Functional & non-functional requirements
Some Requirements Artifacts

- Vision
- Use case model
- Supplementary Specifications
- Glossary
- Data Dictionary
- Business Rules
Use Cases

A use case is a sequence of interactions between a user (actor) and the system that accomplishes something of value to the user.

- Emphasis on needs and/or goals of a specific type of user
- Actors are normally external to system
  - Actor need not be human
  - SuD can be considered an actor in a scenario
Actors and Stakeholders

• An actor is anything with behavior.
  – Primary actor
  – Supporting actors
  – Off-stage actors

• A stakeholder is anyone (or any entity) with an interest in behavior of the use case
  – Primary actor
  – Off-stage actors
  – Others
Writing Use Cases

• Black box use cases
• Levels of formality:
  – Brief
  – Casual
  – Fully dressed
• Main success scenario
• Alternate scenarios
• UI-free style
Identifying Use Cases

- User goals
- Elementary Business Processes
- Probing up the user’s goal hierarchy until one reaches a broad non-EBP goal
- System boundary
Use Cases for the Cooperative Education Support System