Informatik-Projektentwicklung
– Lecture 6 –

Prof. Dr. Peter Müller

Software Component Technology

Wintersemester 03/04

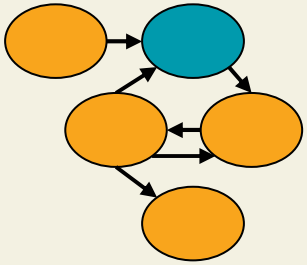
ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Schedule Analysis

The float of an activity is determined by:

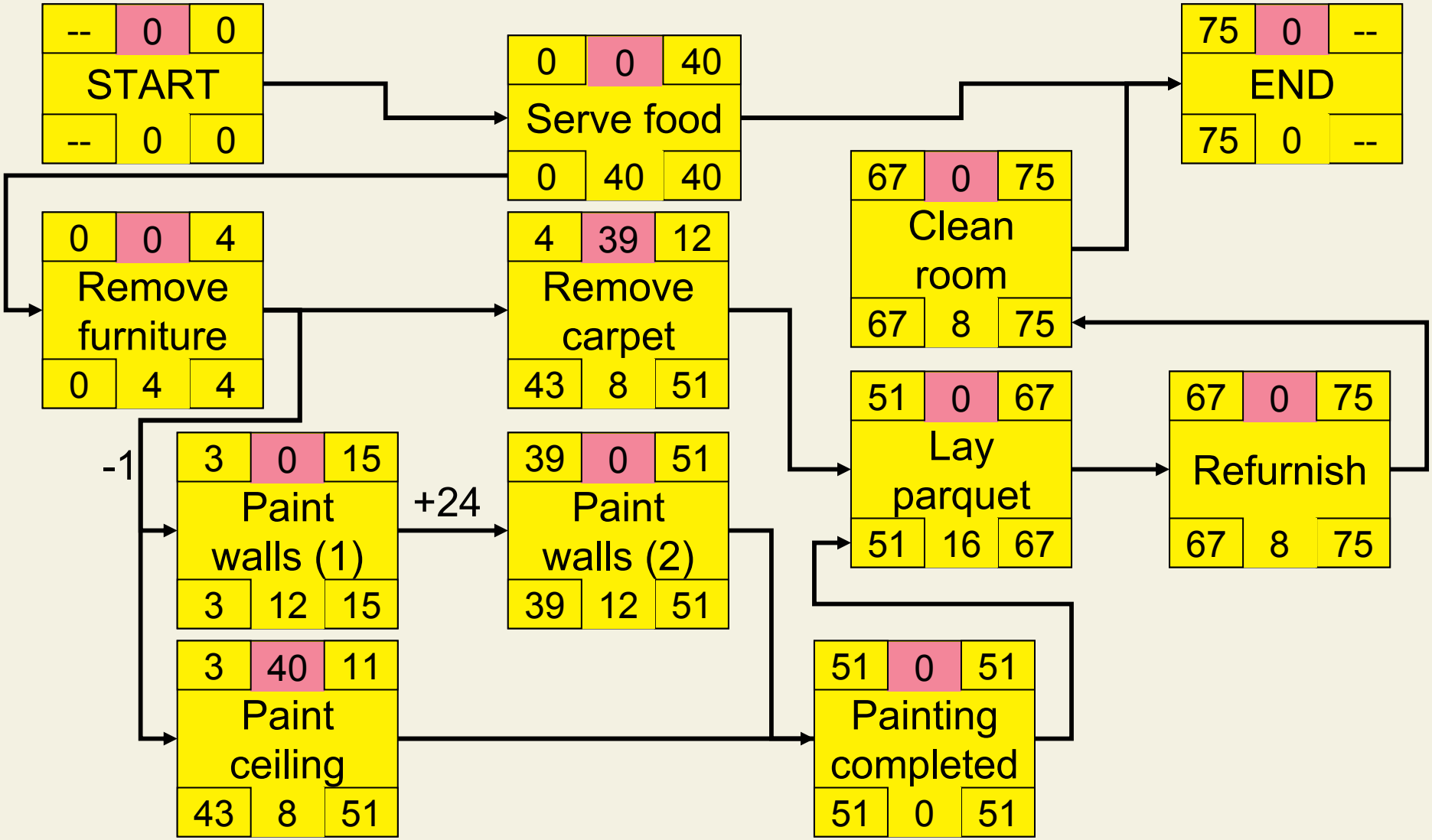
a. The waiting time between tasks
b. Lag
c. The amount of time the activity can be delayed before it delays the critical path
d. The amount of time the activity can be delayed before it delays one of its direct successor activities

Float



- Definition:
  The amount of time that an activity may be delayed from its early start without delaying the project finish date
- Float = LF – EF = LS – ES
- Interpretation
  - Float > 0: Time is available
  - Float = 0: Situation is critical
  - Float < 0: Project is behind
- Sometimes called *Total Float*, *Slack*, or *Total Slack*

Float Example

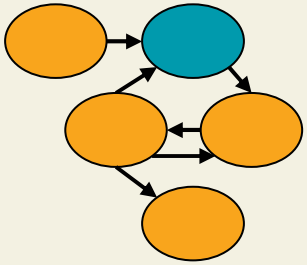


Schedule Analysis

The critical path in a schedule network is the path that:

a. Takes the longest time to complete
b. Must be done before any other tasks
c. Allows some flexibility in scheduling start time
d. Is not affected by schedule slippage

Critical Path

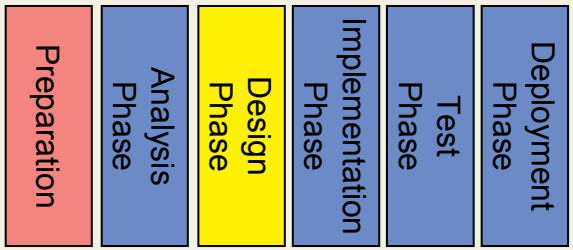


- Definition:
  The series of activities that determines the duration of the project (the longest path through the network)
- Sum of float on critical path is zero (or negative)
- Critical path is important
  - To shorten project duration
  - To focus progress control
  - To identify schedule risks
- There can be several critical paths in a project

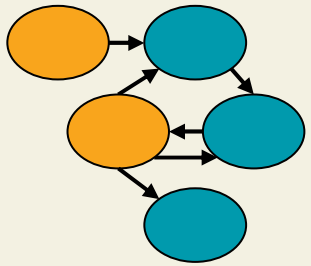
Agenda for Today

6. Design Phase and Procurement Management

6.1 Design Phase



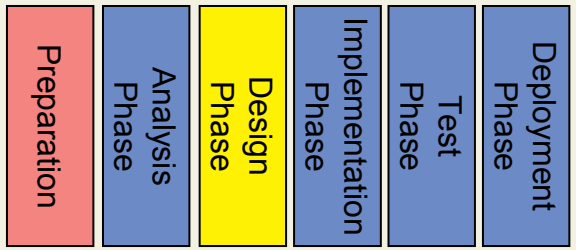
6.2 Procurement Management



6. Design Phase and Procurement Mgmt.

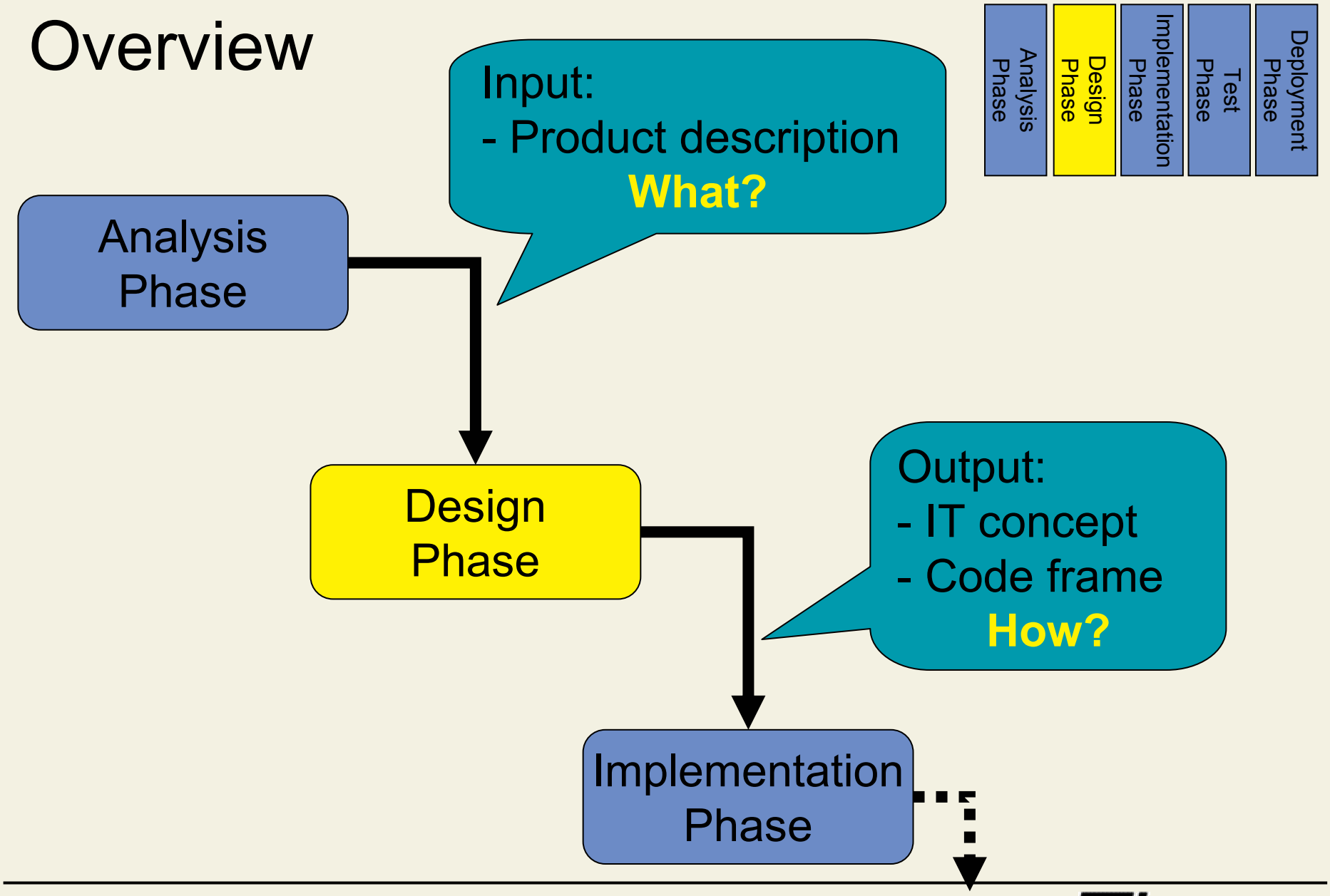
6.1 Design Phase

- Architectural Design
- Detailed Design

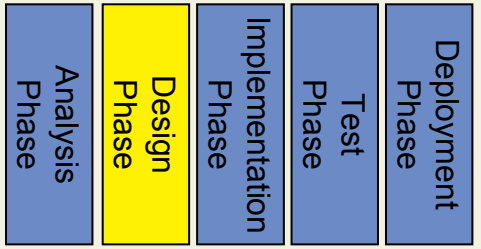


6.2 Procurement Management

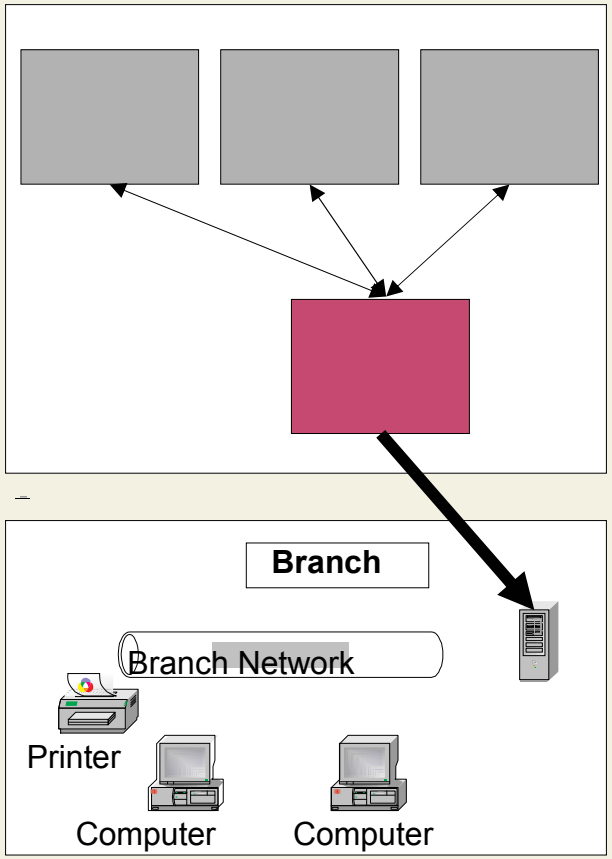
Overview



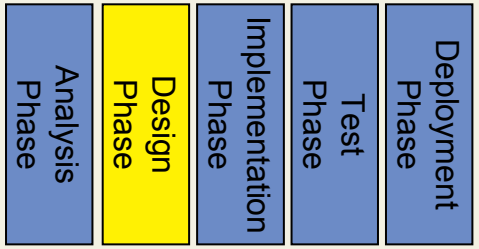
Key Issues



- IT Systems need a clear structure
  - Many people must be able to understand it
  - Development is done by several individuals, teams, subprojects, or companies
  - Only local effects of extensions and adaptations
- Developments have to fit into existing IT landscape
  - Interfaces and dependencies

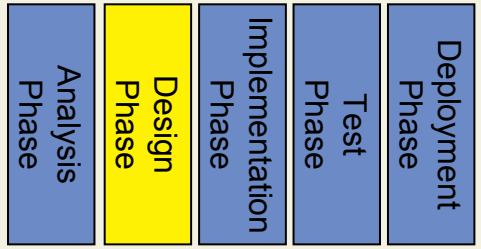


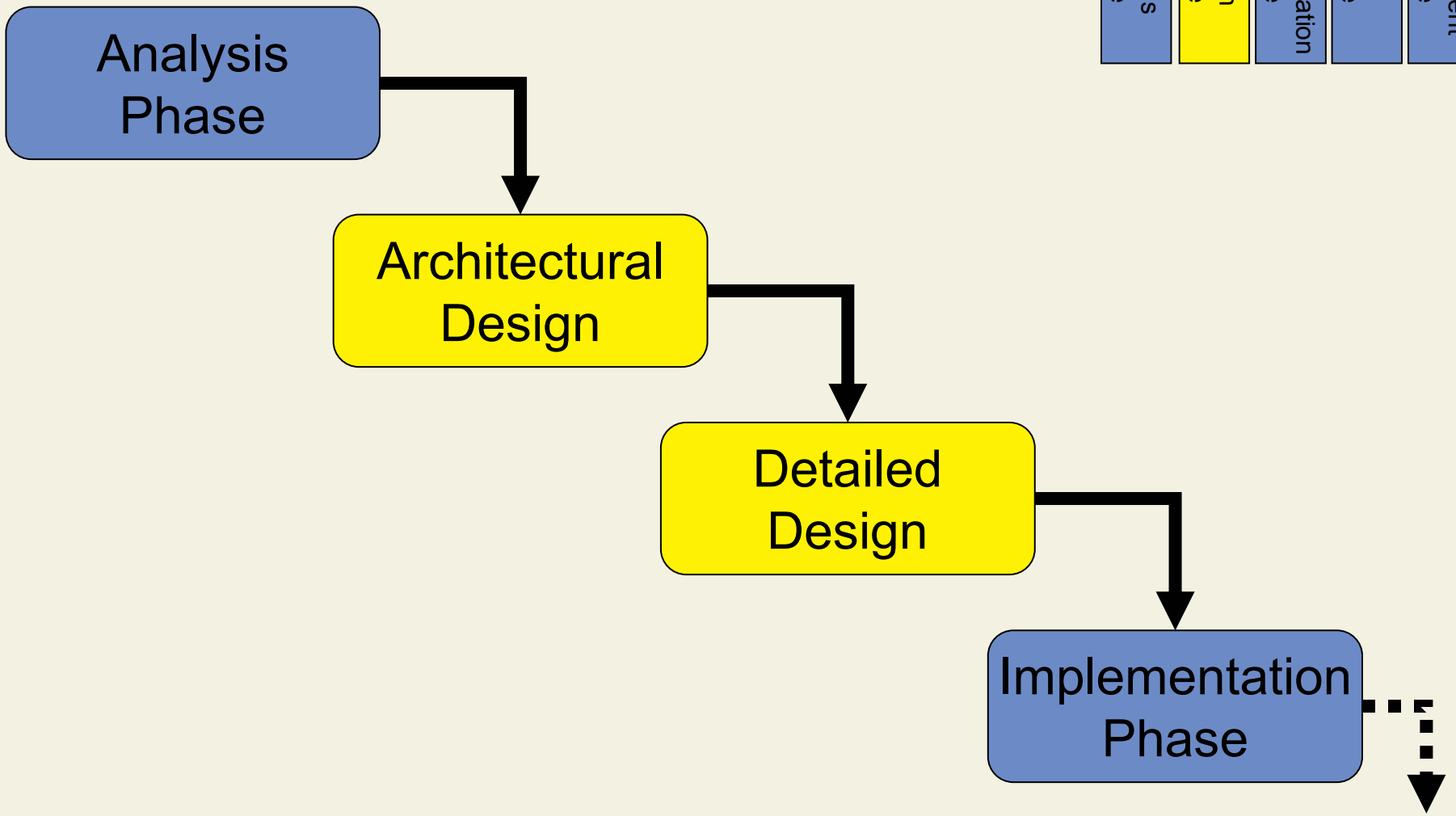
Basic Design Principles



- Decomposition
  - Decomposition of system into components
  - Interaction among components (interfaces and protocols)
  - Distribution of components (on hardware and geographical)
- Abstraction
  - Understanding the big picture by omitting details
  - Understanding details by omitting the rest
  - Systematic refinement

Design Subphases

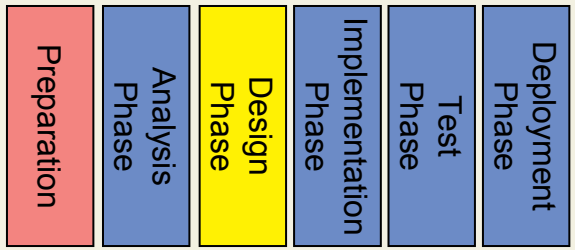




6. Design Phase and Procurement Mgmt.

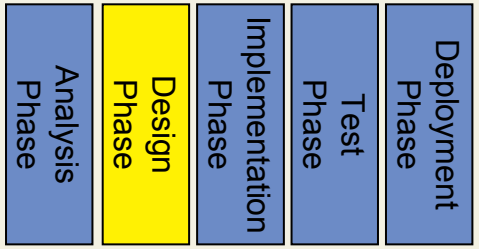
6.1 Design Phase

- **Architectural Design**
- Detailed Design



6.2 Procurement Management

Software Architecture

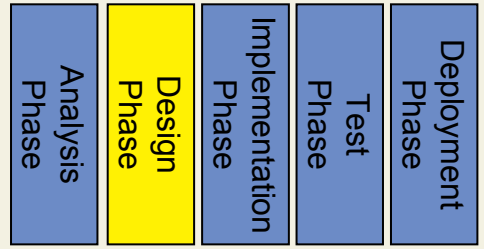


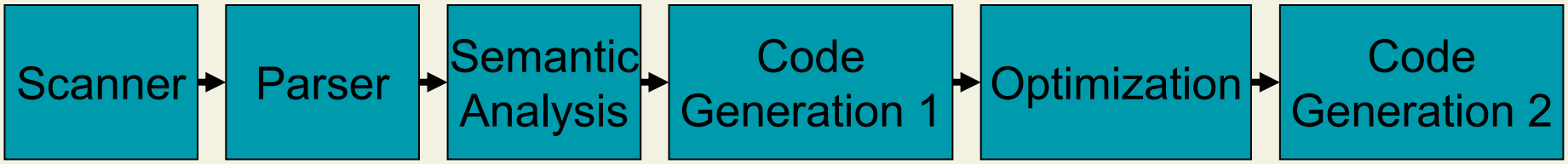
- Definition:
  The architecture of a software system defines that system in terms of computational components and interactions among those components.

  [Shaw, Garlan: Software Architecture]

- Components: Clients and servers, databases, filters, layers in a hierarchical system, etc.
- Interactions: Procedure call, shared variable access, client-server protocols, event multicast, etc.

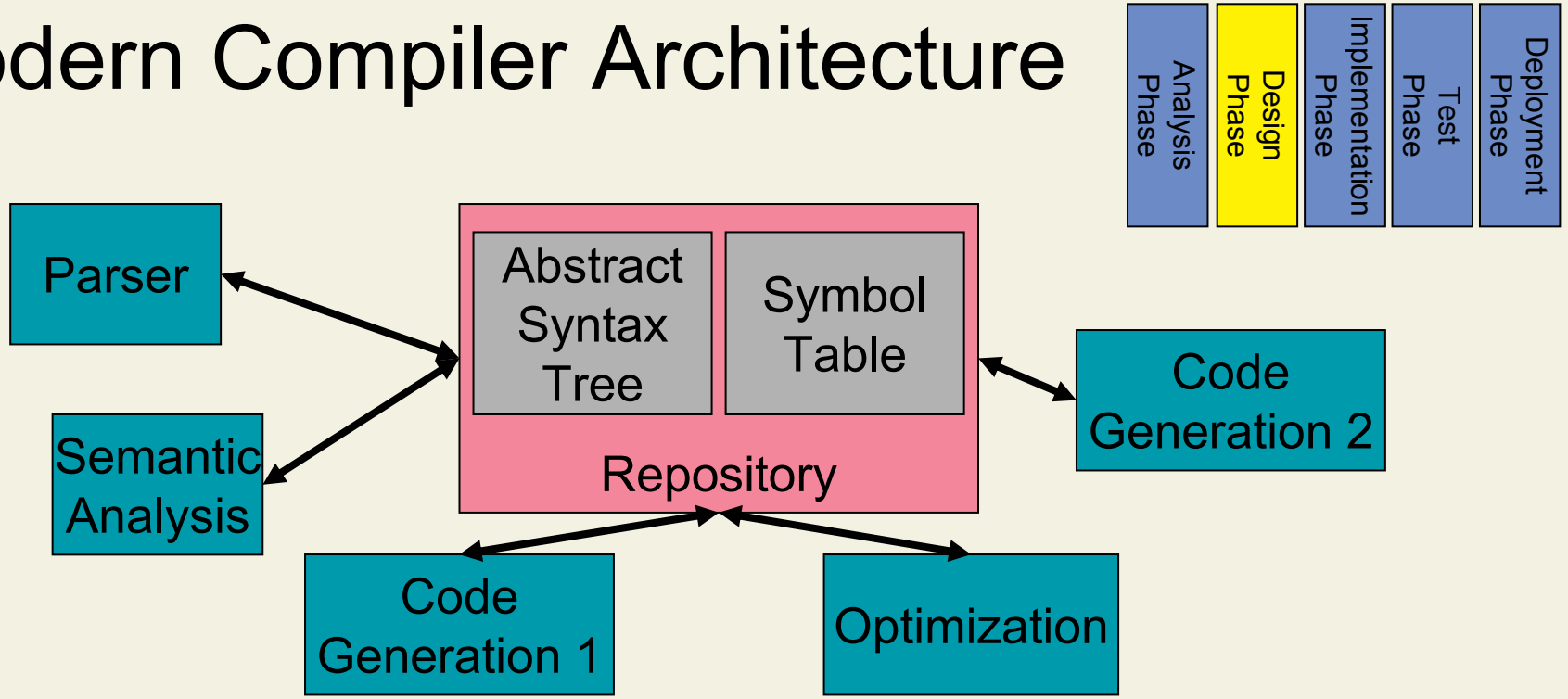
Classical Compiler Architecture





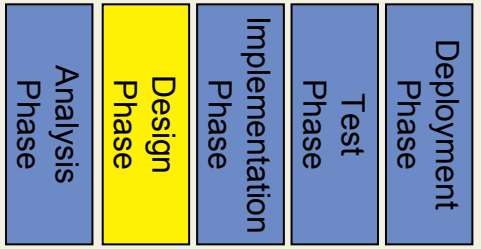
- Pipe-and-filter architecture
- Each component
  - Reads output of predecessor
  - Performs operation
  - Writes output for successor

Modern Compiler Architecture

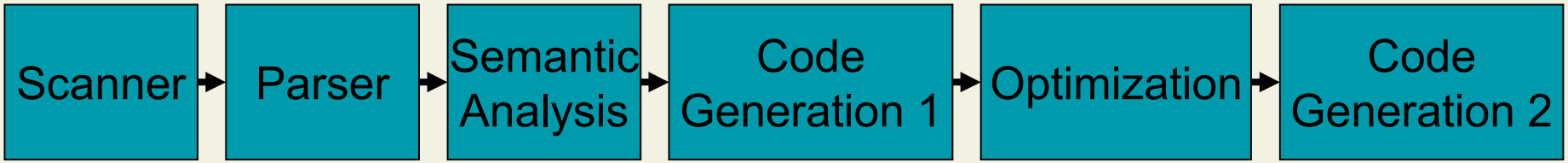


- Repository architecture
- Two kinds of components
  - Central data structure (repository)
  - Computation components

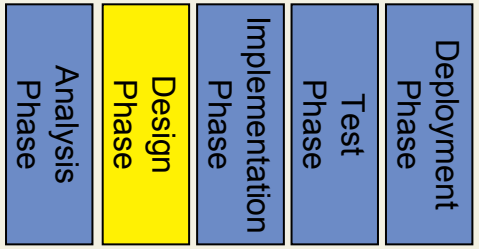
Components



- Components range from whole applications to single classes or modules
- High cohesion within one component
  - Logical relation, common abstraction
- Low coupling between components
  - Small and simple interfaces
  - Black-box components

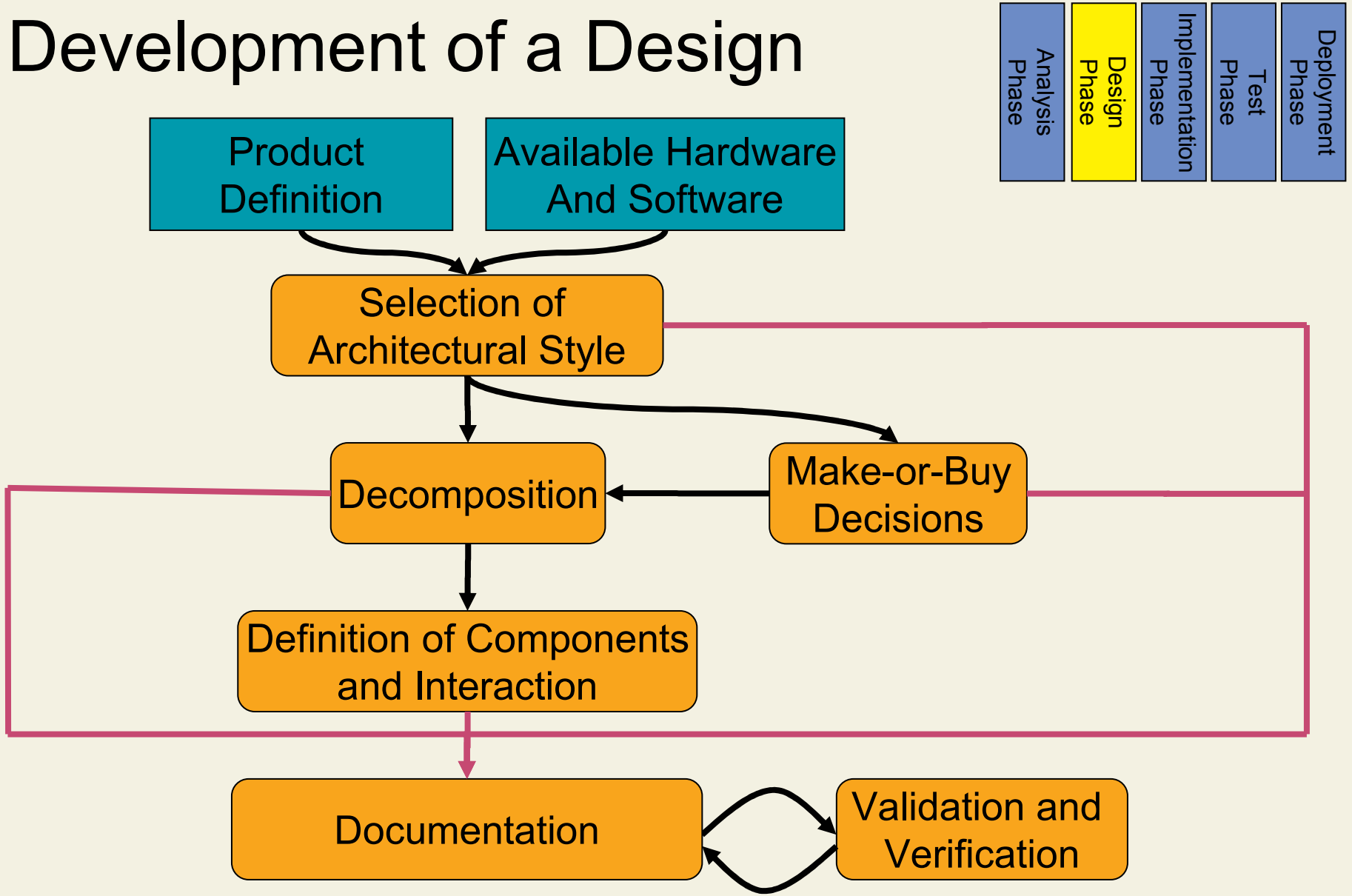


Interfaces

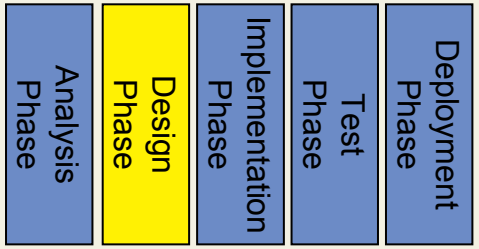


- Kinds of interfaces
  - Internal interfaces between components of the system
  - External interfaces to other applications (within the organization or external)
- Interfaces require communication
  - Both sides of an interface must follow the specification
  - Each change must be discussed and agreed upon
- Guidelines
  - Fix interfaces early in the design and keep them stable
  - Use standards wherever possible

Development of a Design

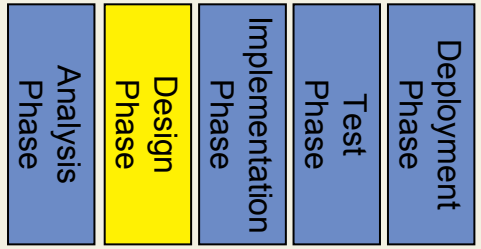


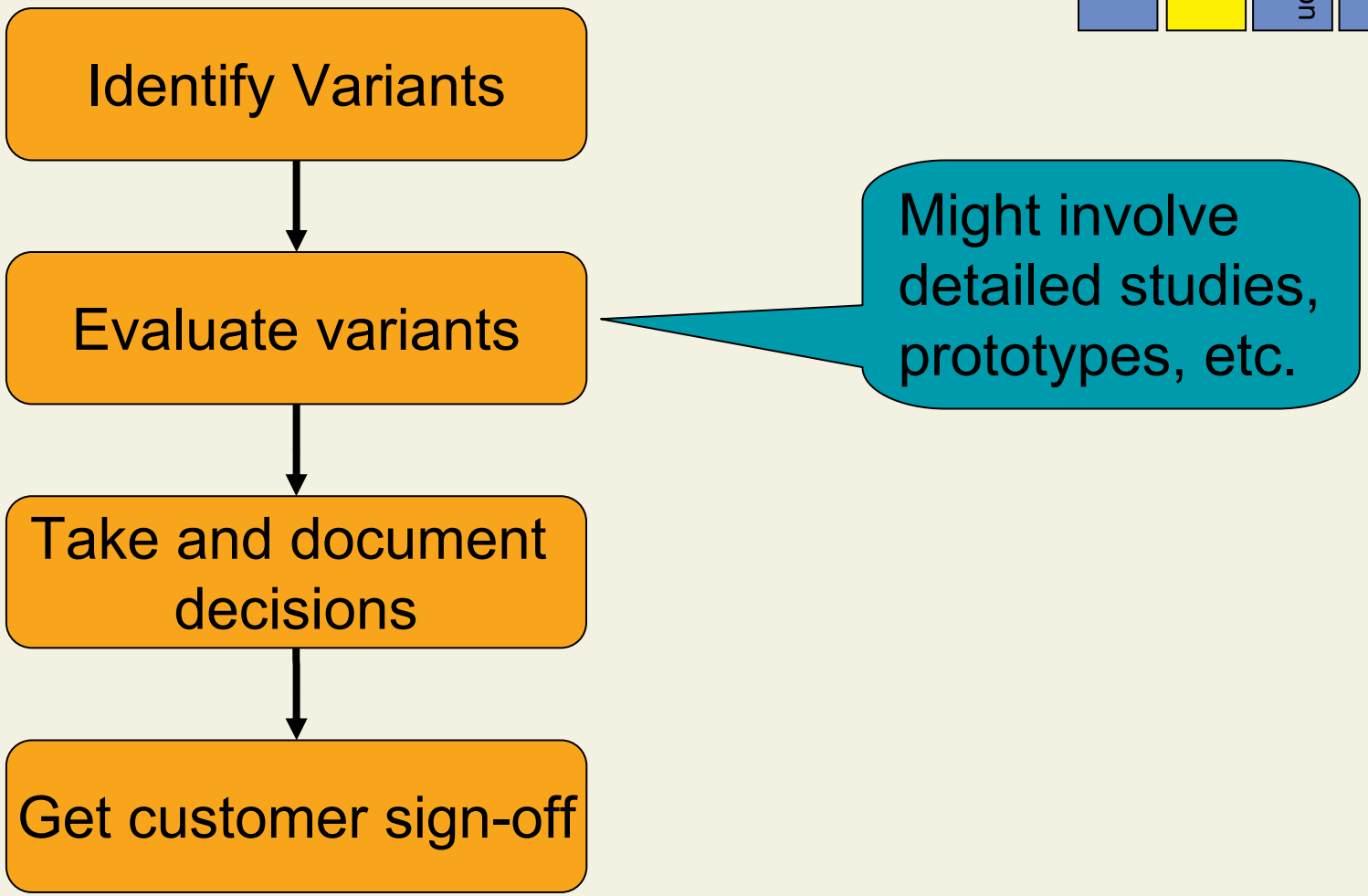
Variants



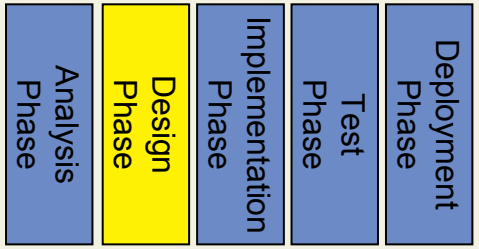
- Possible alternatives
  - Implementing a new system or adapting an old one
  - Building a system or buying a commercial product
  - Centralized or distributed architecture
  - Using a new or an established technology
- Evaluation of variants
  - User requirements (mandatory or nice-to-have)
  - Cost (including development and operation costs)
  - Risk (e.g., missing experience, market trends)
  - Guidelines of the company

Variant Selection





IT Concept

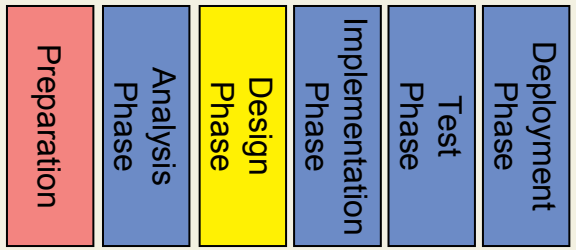


- Main deliverable of architectural design
- Typical Structure
  1. Overview
     - Goals, environment, rejected variants
  2. Solution
     - Architectural style, components, interfaces
  3. Aspects
     - Error handling, security, user interaction
  4. Requirements
     - Software, hardware, environment

6. Design Phase and Procurement Mgmt.

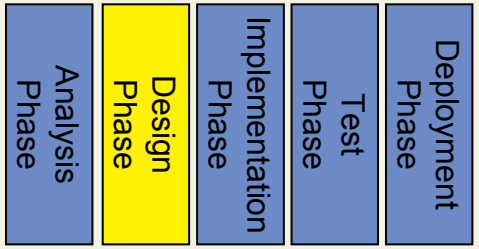
6.1 Design Phase

- Architectural Design
- **Detailed Design**



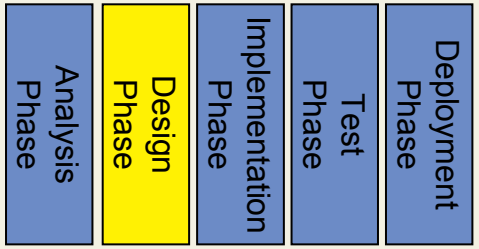
6.2 Procurement Management

Detailed Design

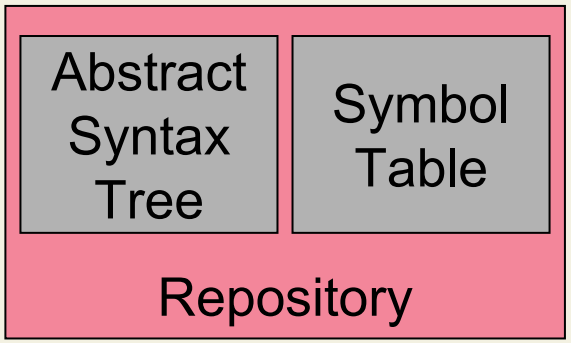


- Mapping components and interfaces to features of the implementation language
  - Class diagrams, module structures
  - Interface specifications (formats, types, etc.)
  - Database schemes
  - Layout of GUI
- Developing a code frame for all components
  - Often generated from design (e.g., from UML diagrams)

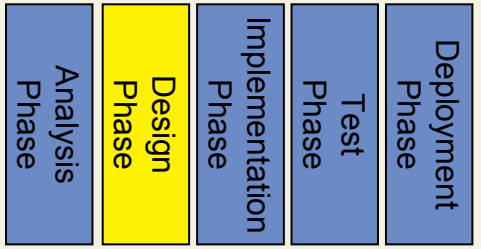
Refinement



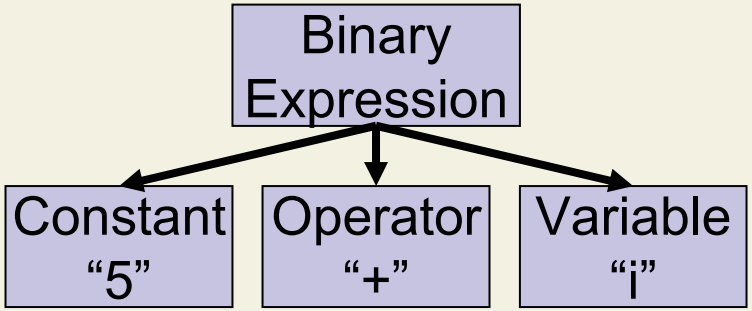
- Elaborate the details of components
  - Continue decomposition
  - Specify components in more detail
- Rule of thumb: Refine until components can be implemented by one person in 1 or 2 weeks

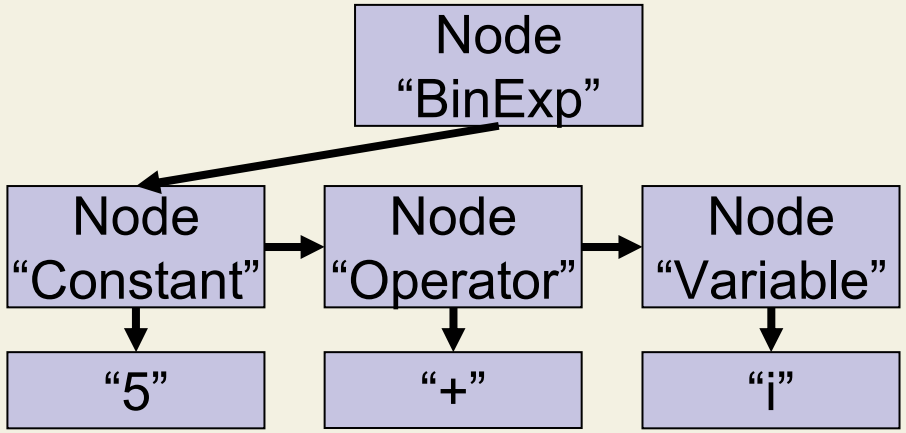


Example

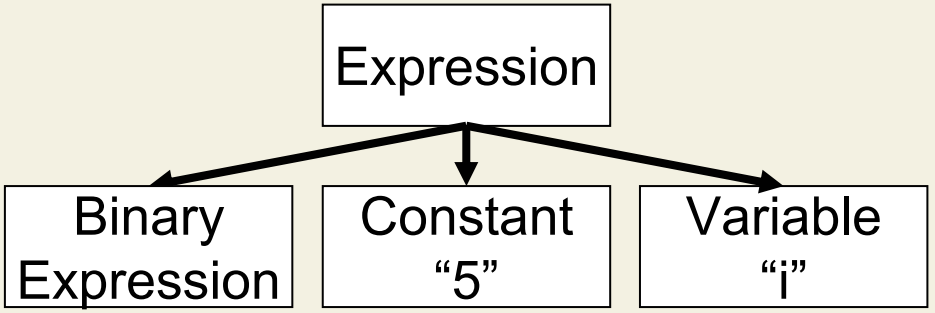


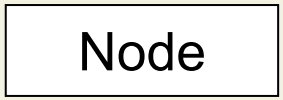
Data Structure:



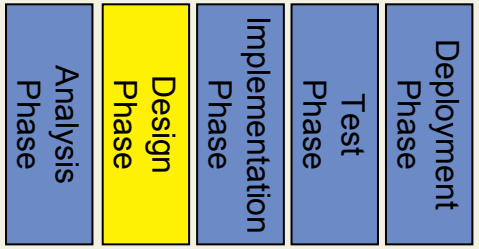


Class Hierarchy:



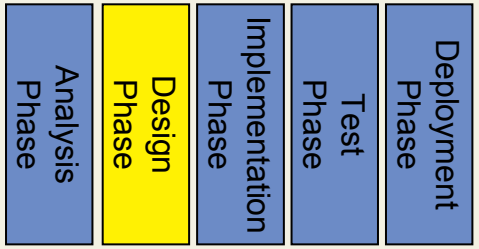


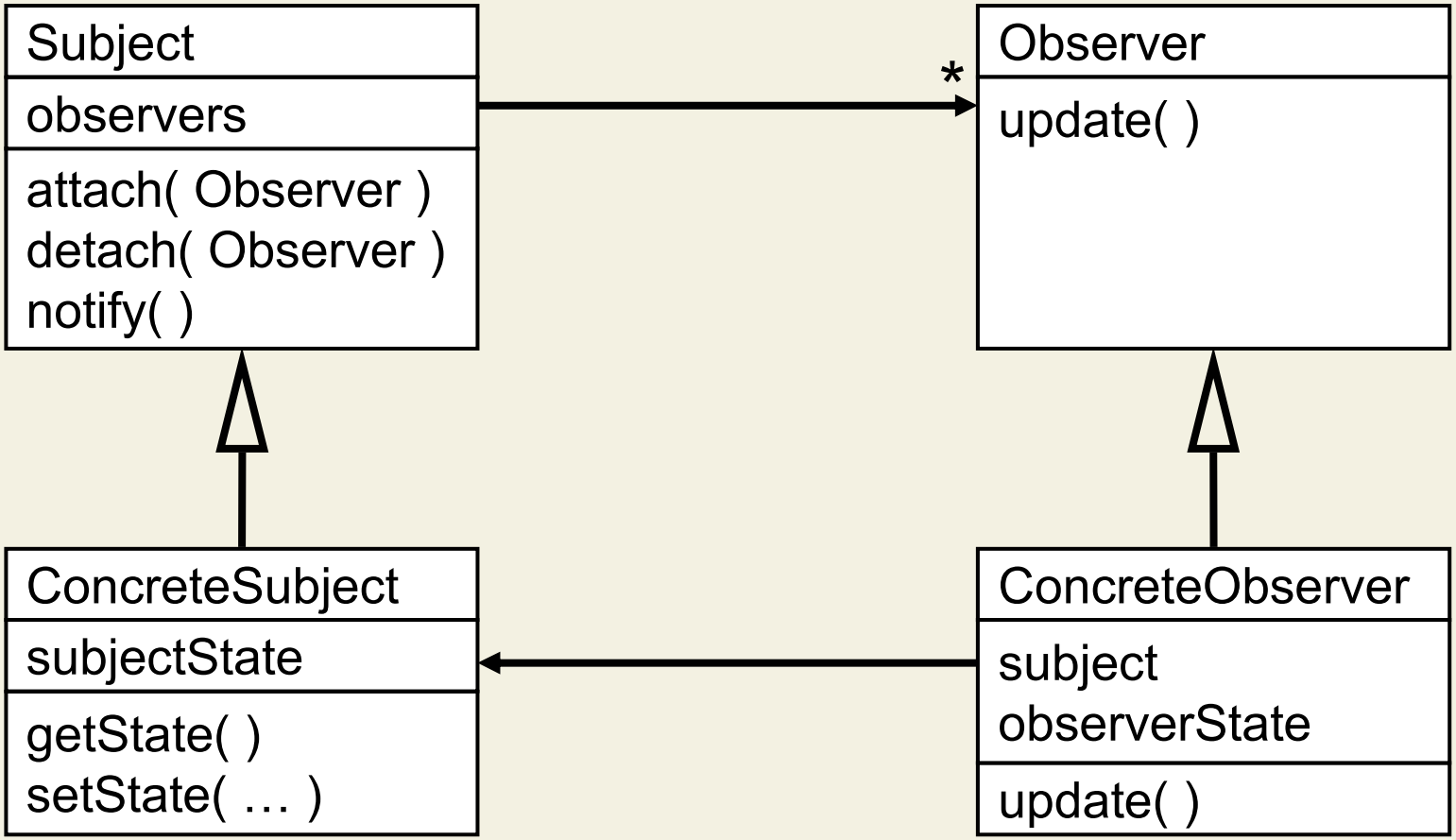
Design Patterns



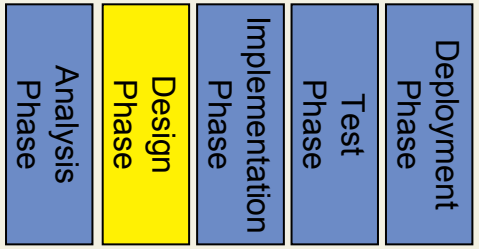
- Definition:
  A description of an object-oriented design technique, which names, abstracts and identifies aspects of a design structure that are useful for creating an object-oriented design. The design pattern identifies classes and instances, their roles, collaborations, and responsibilities.
- Design patterns facilitate
  - Reuse of designs
  - Communication about designs

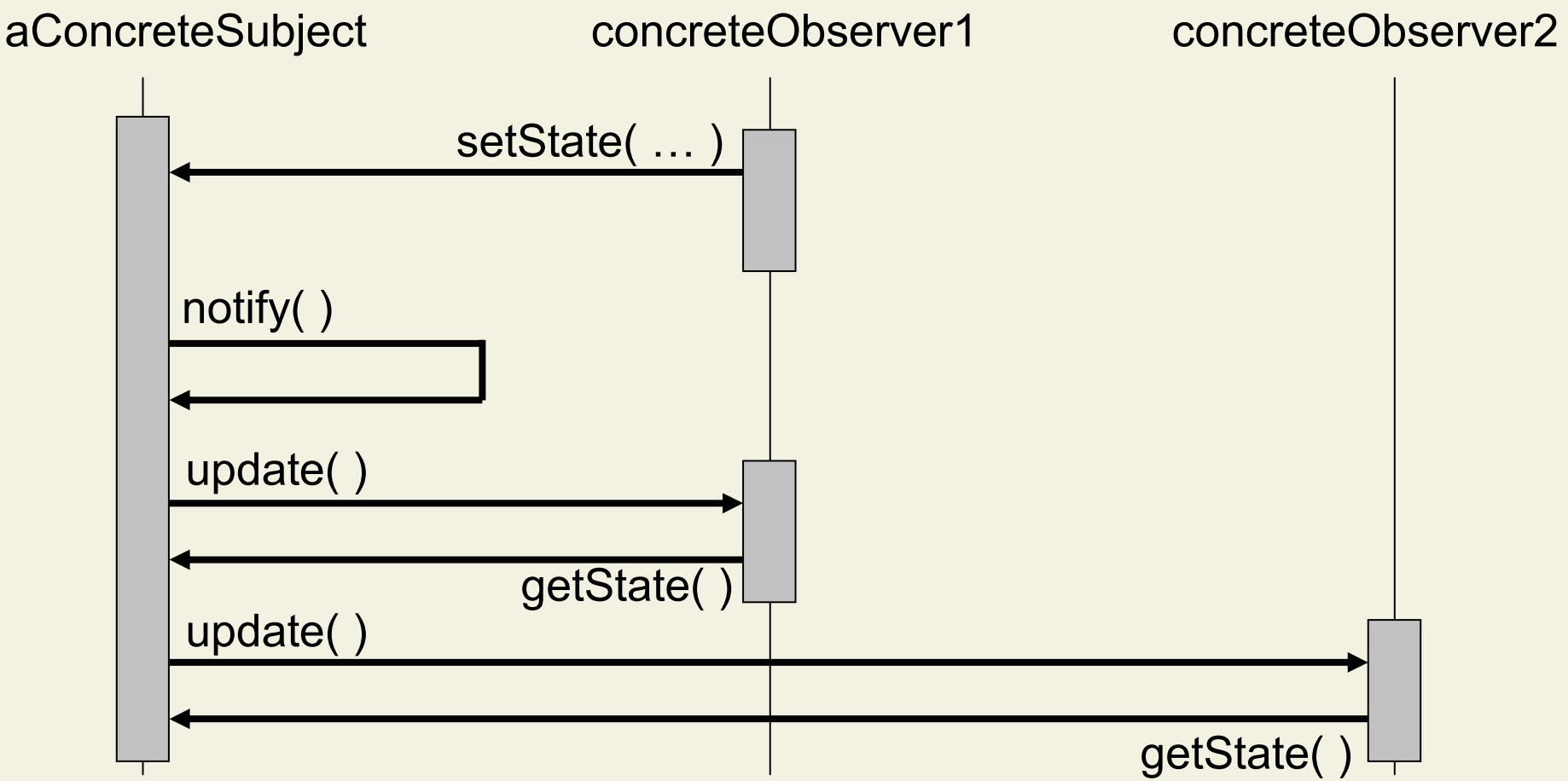
Observer Pattern



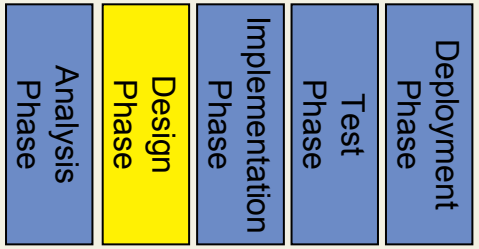


Collaborations



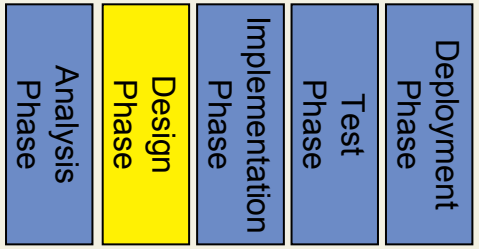


Programming Language



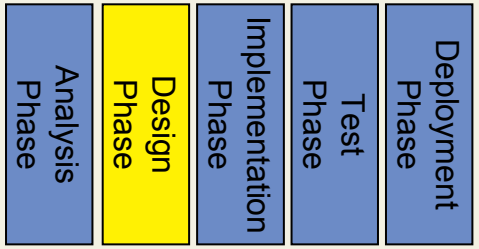
- Implementation languages are selected at the end of the detailed design phase
- Non-technical factors
  - Experience, available knowledge
  - IT landscape, company guidelines
  - Reuse, availability of libraries
  - Tool support
- Technical factors
  - Expressiveness
  - Performance
  - Simplicity

Seamless Development



- Object-orientation uses one method and notation from analysis to implementation
- Abstraction and refinement are expressed by subtyping and inheritance
- Advantages
  - No breaks in methodology, methods, tools, etc.
  - Direct mapping of analysis to code facilitates communication between users, customers, business analysts, designers, and programmers
  - Single framework simplifies backward adjustments

Design Phase: Summary



- Purpose
  - To define how a system meets the specified requirements
  - To decompose a system into manageable components and to specify their functionality and interaction
- Main Deliverables
  - IT concept, documented code frame
- Main actors
  - Architects, designers
- Tools and techniques
  - Decomposition
  - Architectural styles and design patterns

Practice Projects: Phase 2

- Deliverables
  - IT concept and detailed design
  - Detailed schedule with planned and actual efforts
  - Presentation: Design and status

- Timeline
  - Deadline for documents: Thursday, 11.12.03, 15:00
  - Presentations Monday, 15.12.03

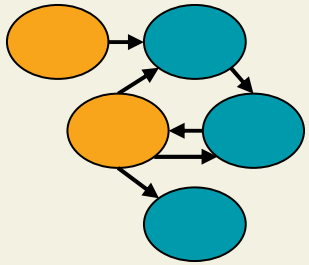
Checklist

- Description of software architecture
  - Component diagram plus textual description
  - Components and interfaces
- Documentation of design decisions
  - Descriptions of alternatives
  - Make-or-buy decisions
- Layout of web pages
- Technology / programming language for each component and interface
- Consult your supervisor!

6. Design Phase and Procurement Mgmt.

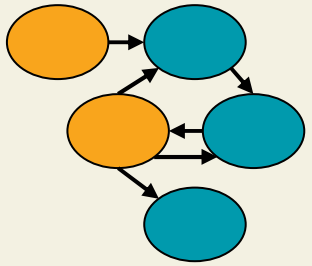
6.1 Design Phase

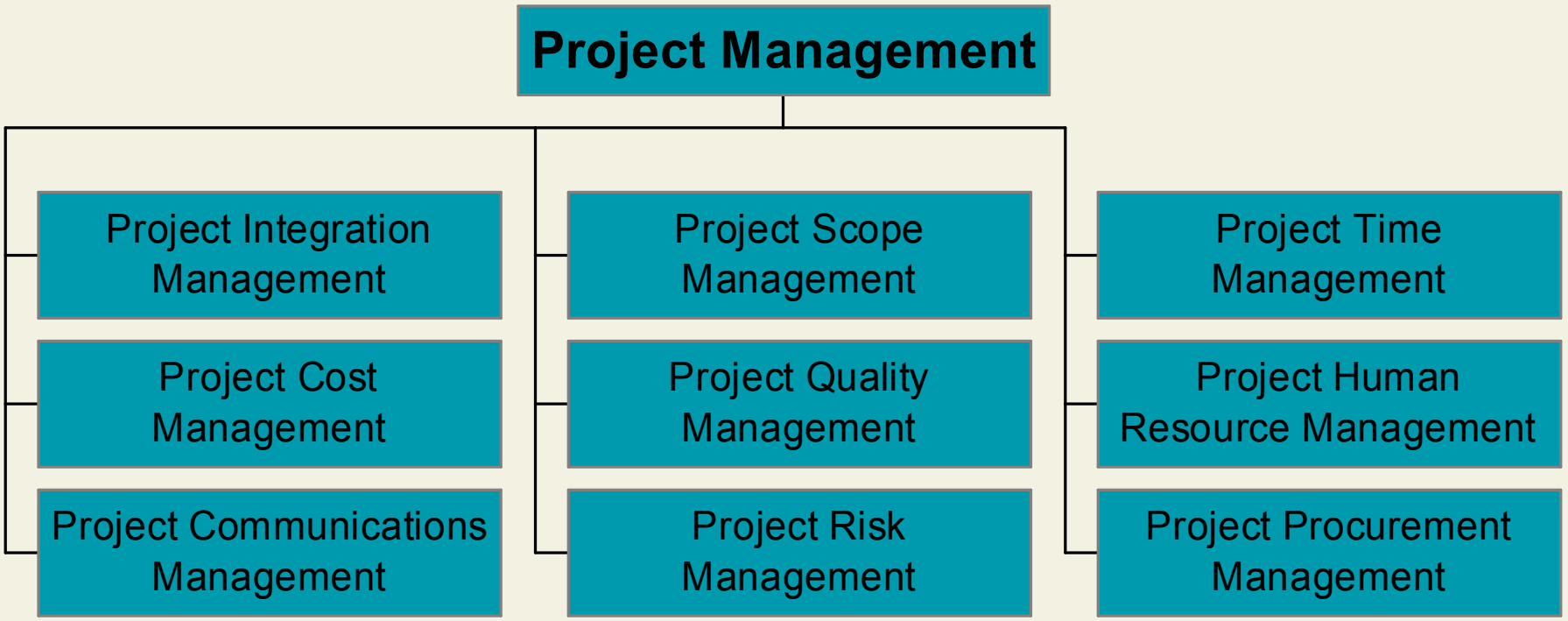
6.2 Procurement Management



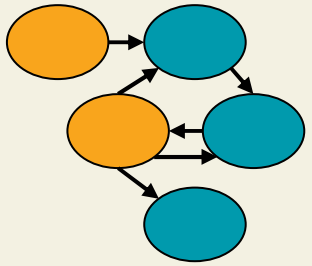
- Procurement Planning
- Solicitation Processes
- Contract Administration and Closeout

Aspects of Project Management





Overview



Planning

Procurement Planning → Solicitation Planning

Execution

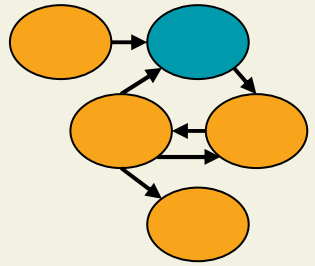
Solicitation

Source Selection → Contract Administration

Closing

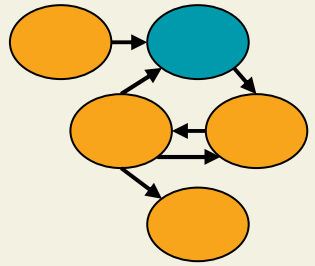
Contract Closeout

Make-or-Buy Decisions



- Each component should be analyzed to determine whether it can be produced cost effectively by the performing organization
- Analysis should include direct and indirect costs
- Analysis must reflect
  - The immediate needs of the project
  - The perspective of the performing organization
- Often a decision between flexibility and cost

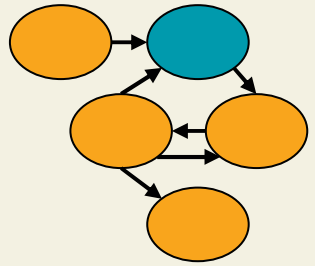
Contracts



- Definition:
  An agreement that establishes an enforceable legal relationship between two parties;
  a mutual exchange of promises.

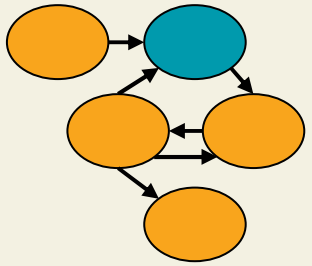
- A good contract should
  - Share risks fairly
  - Motivate each party
  - Balance the interests of the parties involved
  - Prevent surprises

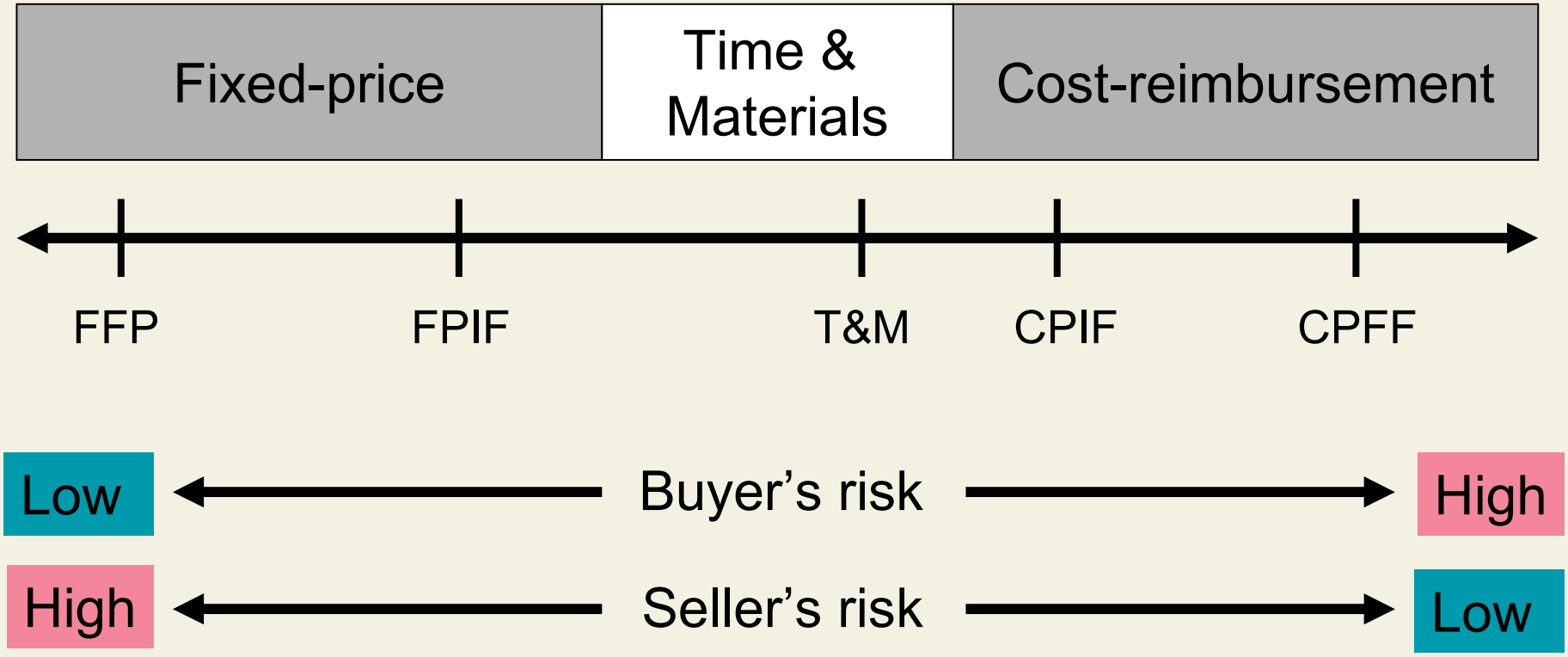
Major Contract Types



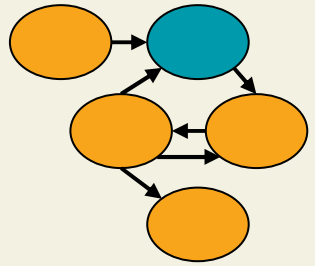
- Fixed-price contracts
  - Firm-fixed-price
  - Fixed-price-incentive fee (incentive if defined performance criteria are met)
- Cost-reimbursable contracts
  - Cost-plus-fixed fee (fixed amount of profit)
  - Cost-plus-incentive fee
- Time & materials (negotiated price per unit of time plus cost of materials)

Contract Types and Risk



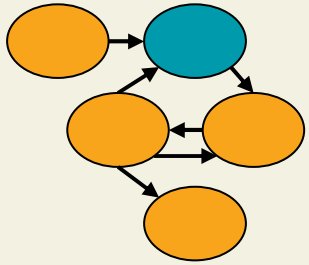


Advantages of Contract Types



- Fixed-price
  - Buyer knows total price at project start
  - Seller has strong incentive to control costs
  - Less work for buyer to manage
- Time & Materials
  - Quick to create
  - Good choice when hiring “bodies”
- Cost-reimbursement
  - Simple scope of work (can start immediately)
  - Generally lower cost (less added for risk)

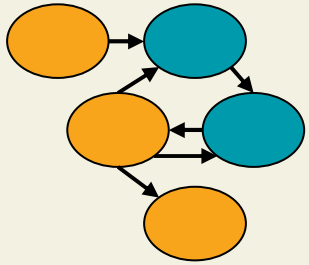
Procurement Planning: Summary



- Purpose
  - To identify which project needs can be best met by procuring products or services outside the project organization
  - To consider whether, how, what, how much, and when to procure

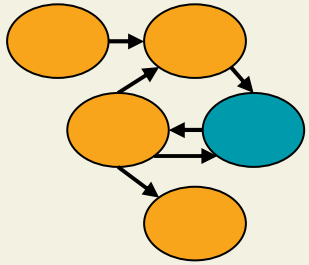
Inputs	Tools & Techniques	Outputs
1. Scope statement 2. Product description 3. Procurement resources 4. Market conditions	1. Make-or-buy analysis 2. Expert judgment 3. Contract type selection	1. Procurement management plan 2. Statement(s) of work

Solicitation Processes: Summary



- Solicitation Planning
  - To prepare the documents needed to support solicitation
- Solicitation
  - To obtain responses (bids and proposals) from prospective sellers
- Source Selection
  - To receive bids or proposals and to apply evaluation criteria to select a provider

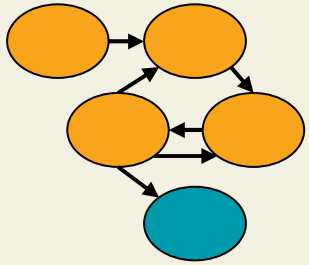
Contract Administration: Summary



- Purpose
  - To ensure that the seller's performance meets contractual requirements

Inputs	Tools & Techniques	Outputs
1. Contract 2. Work results 3. Change requests 4. Seller invoices	1. Contract change control system 2. Performance reporting 3. Payment system	1. Correspondence 2. Contract changes 3. Payment requests

Contract Closeout: Summary



- Purpose
  - To verify products (was all work completed correctly and satisfactorily?)
  - To update records to reflect final results and to archive such information for future use

Inputs	Tools & Techniques	Outputs
1. Contract documentation	1. Procurement audits	1. Contract file 2. Formal acceptance and closure

Systematics of Processes

	Initiating	Planning	Executing	Controlling	Closing
Integration		Project Plan Dev.	Project Plan Execution	Integr. Change Ctrl	
Scope	Initiation	Scope Planning Scope Definition			
Time		Act. Definition, Act. Sequencing, Schedule Dev.			
Cost		Resource Planning			
Quality					
HR					
Comm.					
Risk					
Procurement		Procurement Pl. Solicitation Pl.	Solicitation Source Sel. Contract Admin.		Contract Closeout