Section 3
High-Level System Design

1. Overview
2. Initial system decomposition
3. Design patterns
4. Refined system decomposition

Section 3.1
High-Level System Design Overview

1. Purpose
2. Work products
3. Breakdown

3.1.1 Purpose of High-Level System Design

- This is a crucial software development phase
- This is where our creativity is used
- This phase is not algorithmic
  - it improves with experience
  - it always benefits from team participation

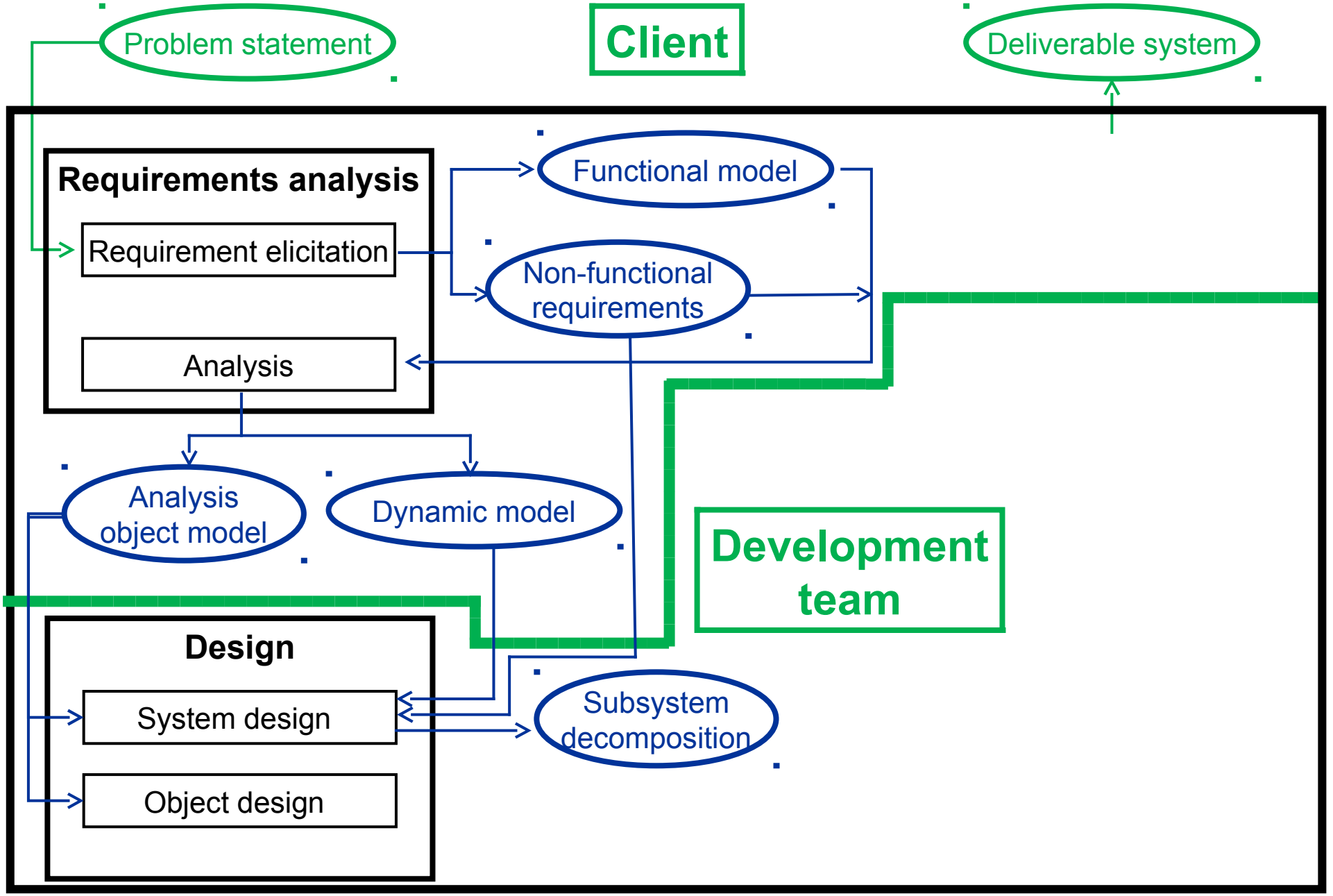
Purpose (cont.)

- Role of development team in high-level system design
  - to transform the analysis model into the system design model
  - to make a first step in mapping the internal structure of the system
    - processes
    - data structures
    - hardware and software components required

Purpose (cont.)

- Input to high-level system design
  - non-functional requirements
  - analysis object model
  - dynamic model
- Output of high-level system design
  - system design model
    - subsystem decomposition
    - system architecture strategies

3.1.2 Work Products



High-Level System Design Tasks

- Main tasks
  - identify design goals
  - design initial system decomposition into subsystems
  - refine subsystem decomposition to meet all design goals

- Identifying system architecture strategies
  - hardware and software mapping
  - persistent data management
  - global control flow
  - access control policy
  - handling of boundary conditions

3.1.3 Breakdown

- High-level system design consists of two parts:
  - initial system decomposition
    - grouping classes into subsystems
  - refined system decomposition
    - addressing design goals
    - deciding on high-level system design strategies

Breakdown (cont.)

- Initial system decomposition
  - input
    - non-functional requirements
    - analysis object model
    - dynamic model
  - output
    - design goals
    - initial subsystem decomposition

Breakdown (cont.)

- Initial system decomposition (cont.)
  - approach
    - determine design goals, criteria from requirements and from client
    - divide the system into maintainable pieces in a way that meets the design goals
  - tools
    - classic system architecture styles
    - established design patterns
    - UML class diagrams, component diagrams, and packages

Breakdown (cont.)

- Refined system decomposition
  - input
    - analysis object model
    - dynamic model
    - design goals
    - initial subsystem decomposition
  - output
    - refined subsystem decomposition
    - system architecture strategies

Breakdown (cont.)

- Refined system decomposition (cont.)
  - approach
    - establish runtime components and nodes
    - identify persistent data, global control flow, access control policies, subsystem services
  - tools
    - UML component diagrams, and deployment diagrams