Section 2
Requirements Analysis

1. Overview
2. Requirements elicitation
3. Analysis

Section 2.1
Requirements Analysis Overview

1. Purpose
2. Work products
3. Breakdown

2.1.1 Purpose of Requirements Analysis

- The *software development life cycle*
  - this is the overall process for the development of large systems
  - it is comprised of several *phases* or *activities*:
    - requirements analysis
    - high-level system design
    - detailed object design
    - implementation
    - testing
- This section focuses on *requirements analysis*

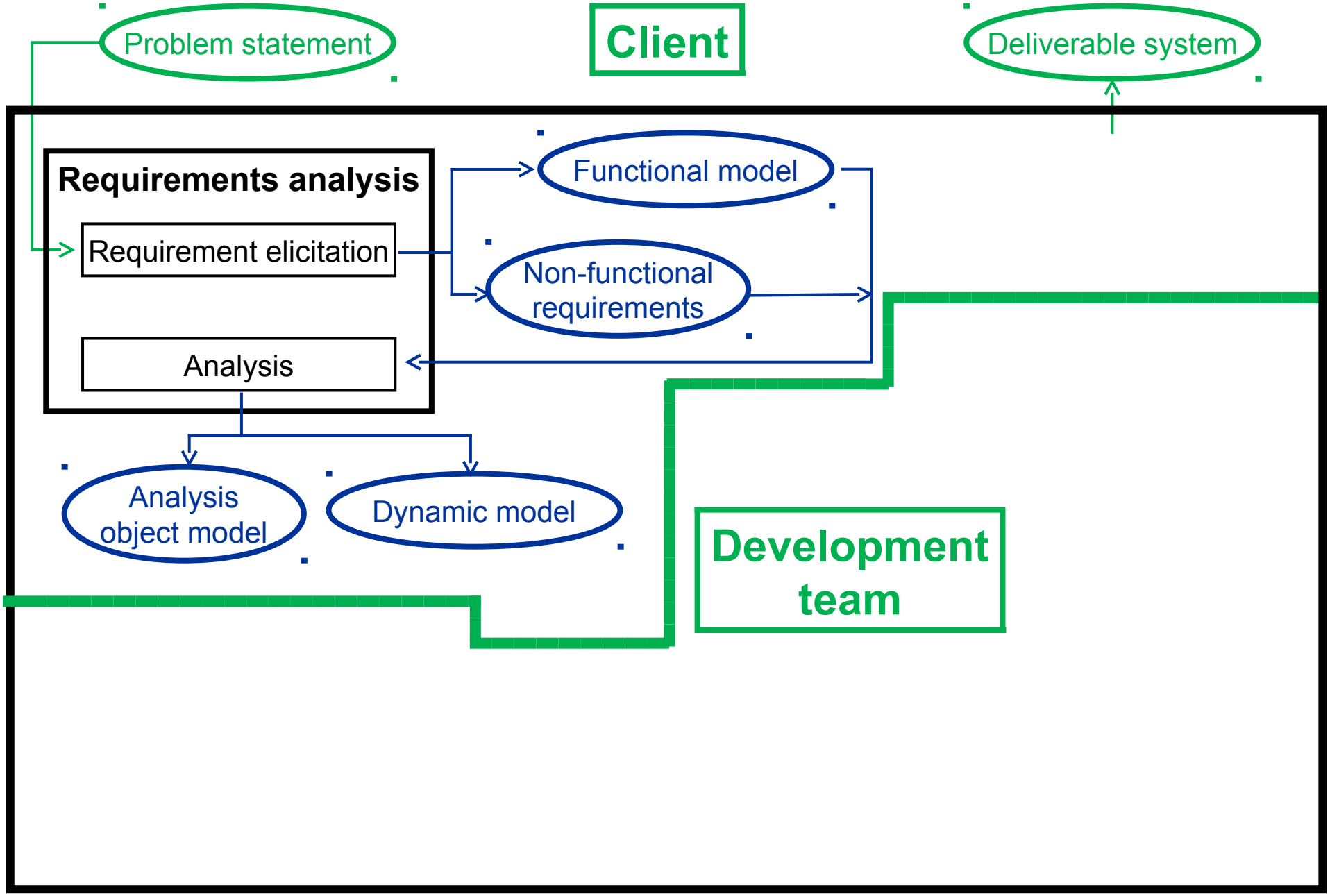
Purpose (cont.)

- Role of development team in requirements analysis
  - to understand the problem to be solved by the new system
  - to model the application domain

- Input to requirements analysis
  - the problem statement

- Output of requirements analysis
  - the functional requirements
  - the functional model
  - the non-functional requirements
  - the dynamic model
  - the analysis object model

2.1.2 Work Products



2.1.3 Breakdown

- Requirements analysis consists of two parts:
  - requirements elicitation
    - gather a detailed and complete set of requirements from the client
  - analysis
    - produce high-level model of the system from the requirements
      - we are still **very** far from coding here
    - focus on system behaviour and object behaviour
      - from the user's point of view
    - model must be understandable by the non-technical client

Breakdown (cont.)

- Requirements elicitation
  - input
    - the problem statement
  - output
    - a *system specification* that:
      - the client understands
      - represents the user's point of view
    - consists of:
      - the functional model (functional requirements, use cases and scenarios)
      - the non-functional requirements

Breakdown (cont.)

- Requirements elicitation (cont.)
  - approach
    - client and user interviews
    - scenario walkthroughs
  - tools
    - use cases
      - use case diagrams
      - table-based descriptions of use cases
    - scenarios

Breakdown (cont.)

- Analysis
  - input
    - the functional model
    - the non-functional requirements
  - output
    - an *unambiguous* high-level model of the system
    - must show:
      - system behaviour (dynamic model)
      - object behaviour and interactions, from the user's point of view

Breakdown (cont.)

- Analysis (cont.)
  - approach
    - analyze use cases to identify user-level objects and behaviour
  - tools
    - state machine diagrams
    - sequence diagrams
    - activity diagrams