V-Model of Software Development

V Model of Software Development

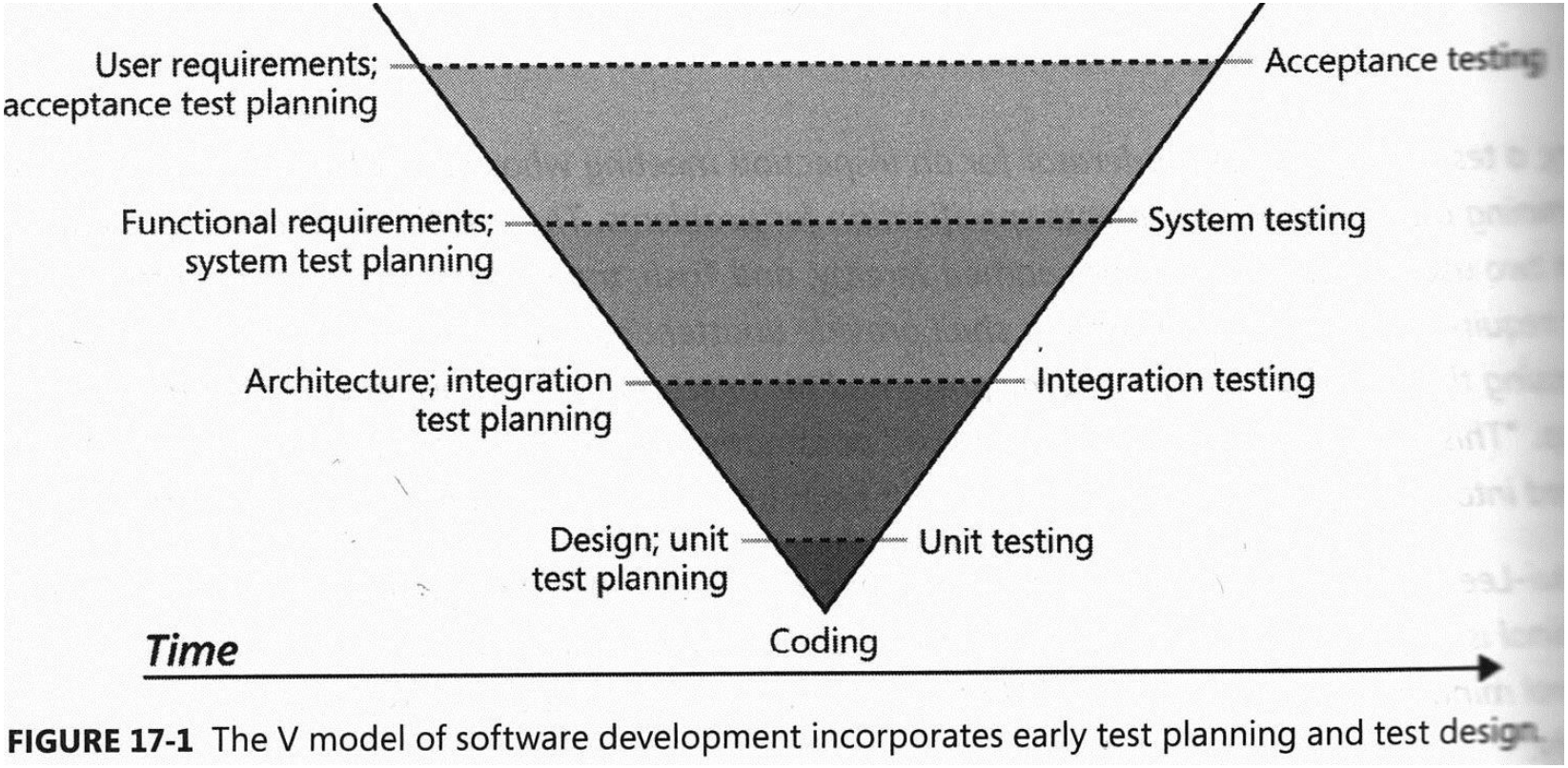

FIGURE 17-1 The V model of software development incorporates early test planning and test design.

Software Requirements, 3rd edition, Karl Wiegers & Joy Beatty, page 330

Development and Testing Derived From Common Source

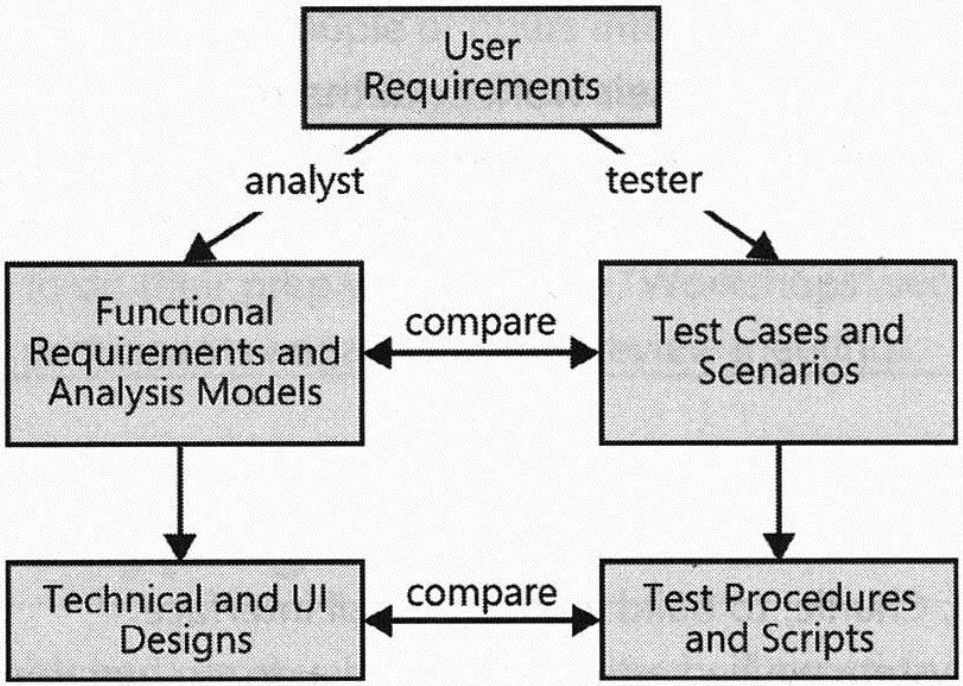

FIGURE 17-5 Development and testing work products are derived from a common source.

Software Requirements, 3rd edition, Karl Wiegers & Joy Beatty, page 344

Conceptual Tests

Conceptual (implementation independent) tests based on the requirements reveal:

- errors
- ambiguities
- omissions

in your requirements and models

Software Requirements, 3rd edition, Karl Wiegers & Joy Beatty, page331

Requirements Development Phases

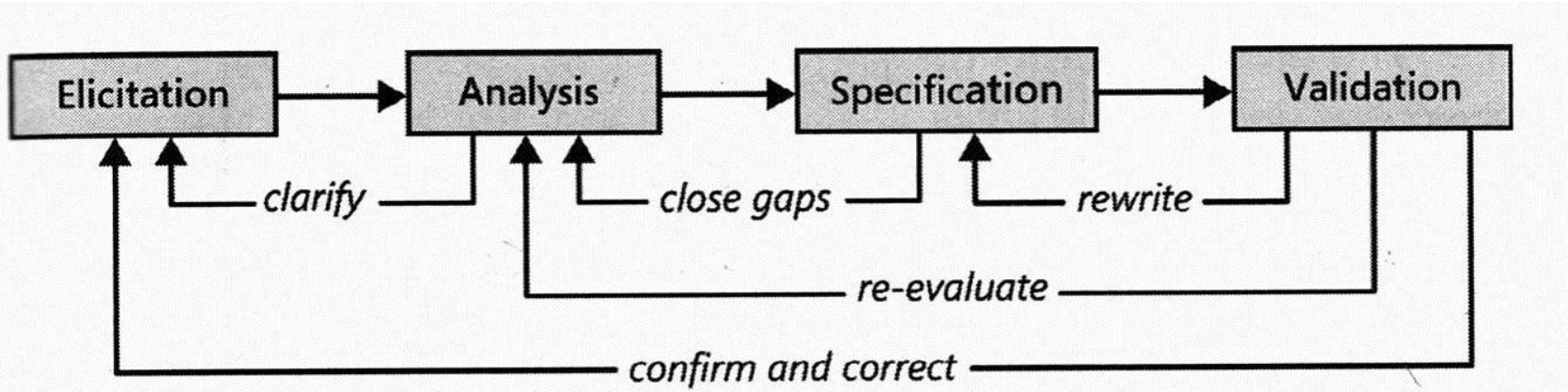

FIGURE 3-1 Requirements development is an iterative process.

Validation, combination or validation and verification

- Validation – Building the right system, satisfies customer's needs
- Verification – Building the system right, using good practices

Software Requirements, 3rd edition, Karl Wiegers & Joy Beatty, page 45

Requirements Development Phases

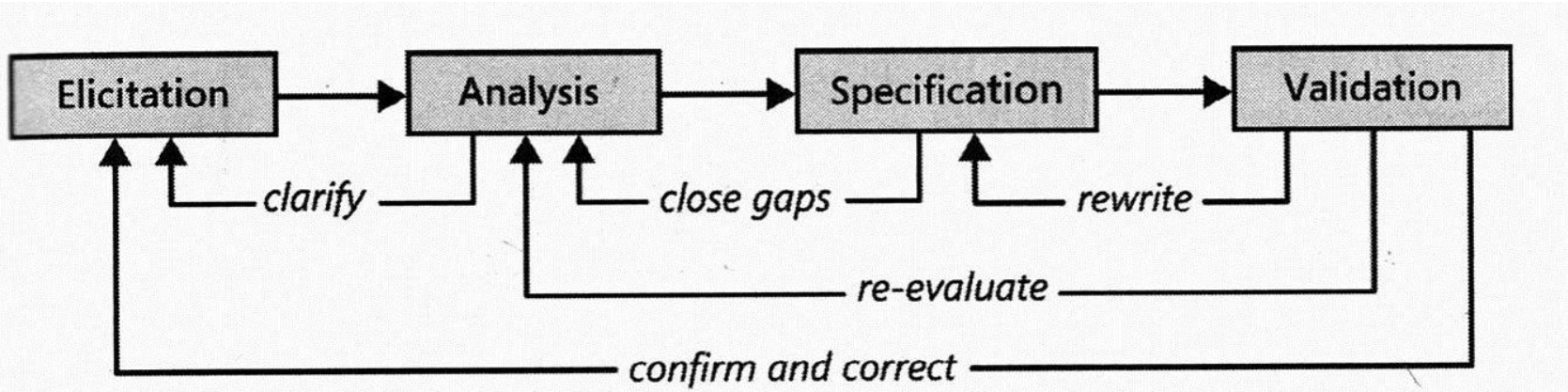

FIGURE 3-1 Requirements development is an iterative process.

Validation, combination or validation and verification

- Validation – Building the right system, satisfies customer's needs
- Verification – Building the system right, using good practices

Software Requirements, 3rd edition, Karl Wiegers & Joy Beatty, page 45

Validating Requirements

Requirements validation activities attempt to ensure that the requirements:

- accurately describe intended system capabilities and properties
- are complete, feasible and verifiable
- are necessary, and the entire set meets the business objectives
- representations are consistent with each other
- provide adequate bases to proceed with design and construction

Software Requirements, 3rd edition, Karl Wiegers & Joy Beatty, page331

Test Coverage

Tests should cover:

- normal flow
- Alternative flows
- exceptions

Software Requirements, 3rd edition, Karl Wiegers & Joy Beatty, page343

Requirements Analysis

Requirements of the system are collected by analyzing the needs of the users. The ideal system is established, without concern for how the software will be designed or built.

A requirements document may be created describing the system's functions, interface, performance, data, security, etc. This can be used by business analysts to communicate their understanding of the system to the users. Methods for gathering requirements may include interviews, questionnaires, document analysis, observation, throw-away prototypes, use case, static and dynamic views of the system.

User Acceptance Test (UAT) - Composed by business users. Performed in a user environment that resembles the production environment. Uses realistic data. Verifies that delivered system meets user's requirements.

Wikipedia – V-Model

System Design

System engineers analyze and understand the business of the proposed system by studying user requirements. They figure out possibilities and techniques by which the user requirements can be implemented. If any of the requirements are not feasible, the user is informed of the issue. A resolution is found and the user requirements are edited accordingly.

A software specification document, containing the system organization, menu structures, data structures, etc. may be generated to serve as a blueprint for the development phase. Example business scenarios, sample windows , reports, entity diagrams and a data dictionary may be created.

System Tests Plans - Composed by client's business team, the whole application is tested for its functionality, interdependency and communication. Verifies that functional and non-functional requirements have been met including load and performance testing, stress testing, and regression testing.

Wikipedia – V-Model

Architecture Design

High-level design, list of all modules, brief functionality of each module, their interface relationships, dependencies, database tables, architecture diagrams, technology details, etc.

Integration Test Plans - Verify that units created and tested independently can coexist and communicate among themselves.

Wikipedia – V-Model

Module Design

Low-level design, system is broken into small units or modules so programmers can begin coding.

Determine:

- Type and size of all database tables elements,
- all interface details with complete API references
- all dependency issues
- error message listings
- complete input and outputs for a module.

Unit Test Plans (UTPs) - Unit is the smallest entity which can independently exist, e.g. a program module. Unit testing verifies that the smallest entity can function correctly when isolated from the rest of the codes/units.

Wikipedia – V-Model