NDSU

Introduction to Formal Specification

Formal Methods of S/W Development

Formal Methods (of SW Dev.) NDSU

- **Formal Methods**

  Rigorous mathematically-based techniques and tools for the specification, development, and verification of software and hardware systems.

- **Software Development**

  The process by which user needs are translated into a software product. This involves translating user needs into software requirements, transforming the software requirements into design, implementing the design in code, and testing the code[1]. (5/5 points)

1. IEEE Std 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology

Development Process

Key Activities

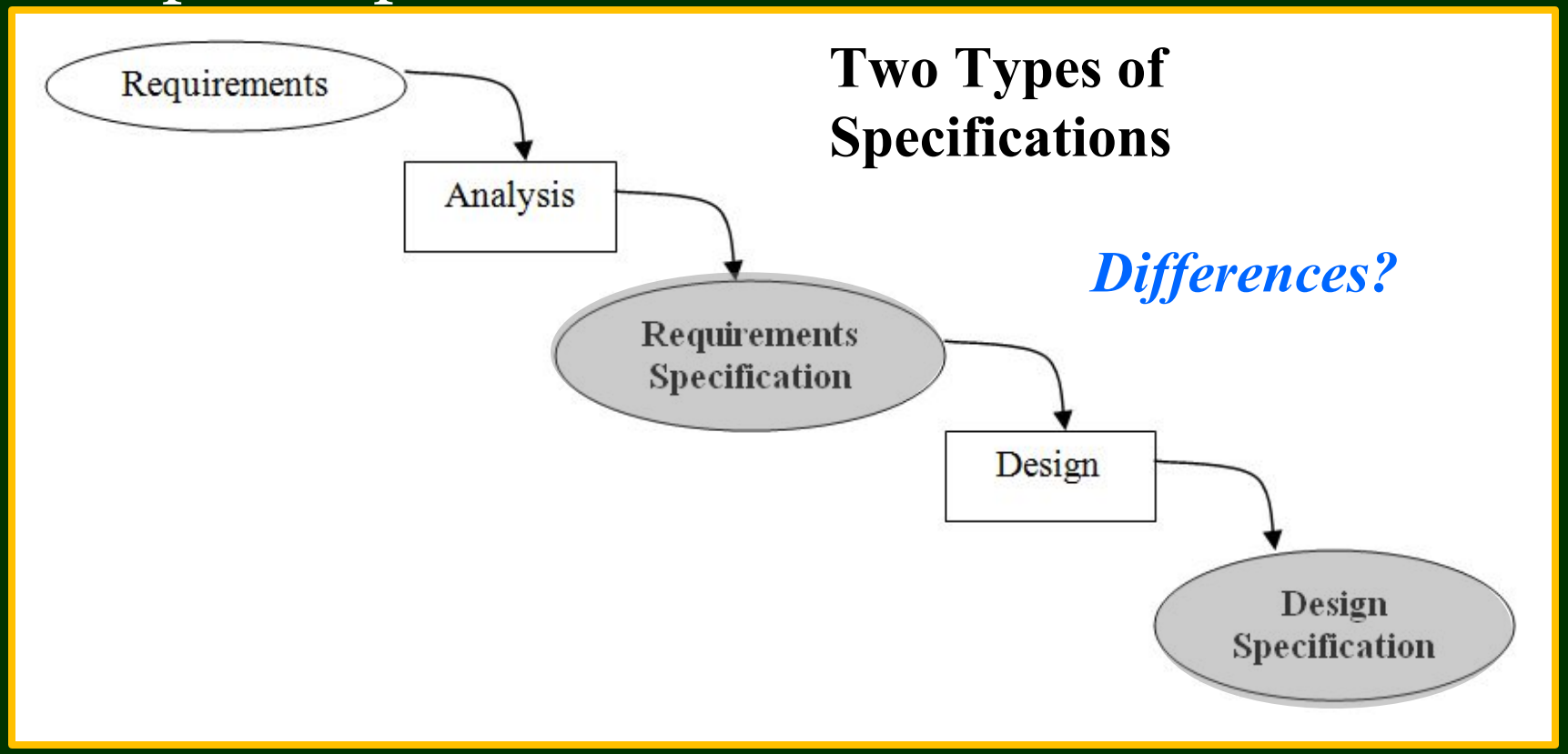
- Requirements Analysis
- Design
- Implementation
- Validation & Verification

Process Models

- Waterfall, V-Model, Spiral Model, Incremental Development

What is a Specification?

- **Specification:** Intermediate product of software development process



Specification impacts Design and Implementation

Also viewed as a…

- Basis for Ensuring Correctness
  - Correctness defined as product satisfies its specification
  - Established by V&V, i.e., impacts testing and verification
- Contractual Agreement
  - Client signs off on the SRS
- Means of Communicating Ideas
  - Provides a high-level description or big picture
  - Acts as a reference point for different stakeholders

Desirable Features (Content)

The contents of a **specification** should be:

- Correct – accurately represents the needs of the user
- Consistent – contains and derives no contradictions
- Complete – covers all possible scenarios and error cases
- Unambiguous (Precise) – provides exact descriptions that have exactly one meaning, neither more nor less.
- Verifiable – allows specification to be checked to determine whether or not it satisfies (meets) predefined criteria

Desirable Features (Content)

Describes software systems and therefore…

- Structural – captures hierarchical and uses relationships
- Behavioral – addresses required functionality and non-functional aspects such as fault tolerance, safety, security

What about the specification language itself?

What are desirable characteristics of a language?

What is the most important characteristic?

Desirable Features (Language) NDSU

- Understandability – purpose of a language is to facilitate communication. For software must handle complexity.

  Features that make a language (more) understandable?

  - Graphical – visual notations such as diagrams
  - Hierarchical – different levels of abstraction
  - Composable – divide and conquer

- Expressive – powerful and meaningful descriptions

- Analyzable – amenable to machine manipulation

Specification Languages

- Specification languages (like all languages) have a syntax and semantics
- Syntax provides a set of symbols and a set of grammatical rules for combining those symbols into sentences; while semantics ascribe meaning to them.
- A specification language may be classified according to its:
  - Foundation – basis upon which it was created
  - Applicability – expressiveness for different system types
  - Style – format of notation or representations used.

Classifications (cont'd)

The *model-oriented* approach to specification is based on mathematical models, and a model is a mathematical representation or abstraction of a physical entity or system.

The model aims to provide a mathematical explanation of the behavior of the physical world, and it is considered suitable if its properties closely match those of the system being modeled

A model will allow predictions of future behavior to be made, and many models are employed in the physical world (e.g., weather forecasting system).

Classifications (cont'd)

- The *axiomatic approach* focuses on the properties that the proposed system is to satisfy, and there is no intention to produce an abstract model of the system. The required properties and behavior of the system are stated in mathematical notation.

- Classifications

- Formal methods may be model oriented or axiomatic oriented. The model-oriented approach includes formal methods such as VDM, Z and B. The axiomatic approach includes the process calculi such as CSP, CCS and the π calculus

Formal Specification

- Formal specification is the use of mathematical notation to precisely describe what properties a system should have, without unduly constraining how to achieve them.
- Engineers that use formal specification techniques are faced with the challenge of developing a "complete" **system description** that is free of implementation bias.
- Using mathematical data types for modeling allows the abstraction away from computer representation, while providing a rich collection of laws for effectively reasoning about how the specified system will behave

Formal Spec. Languages

- A Formal Specification Language (FSL) provides the sound mathematical basis for a formal method.

- **Formal Definition**: An FSL is a triple $< Syn, Sem, Sat >$ where,

  Syn and Sem are sets and

  $Sat \subseteq Syn \times Sem$ is a relation between them.

- Syn is the syntactic domain of the language
  Sem is the semantic domain and
  Sat is its satisfies relation.

Jeannette M. Wing '90, A Specifier's Introduction to Formal Methods

Formal Spec. Languages (cont'd) NDSU

- FSLs provide a notation (syntactic domain), a universe of objects (semantic domain), and a precise rule defining which objects satisfy each specification.
- A specification is a sentence written in terms of the syntax, and an object satisfying a specification is a specificand.

Examples of FSLs

- Z (pronounced "Zed") is based on set theory and first-order predicate logic. Can used in model-oriented . Applies to sequential systems.
- VDM is a *model-oriented approach* and this means that an explicit model of the state of an abstract machine is given, and operations are defined in terms of this state. Operations may act on the system state, taking inputs, and producing outputs as well as a new system state.