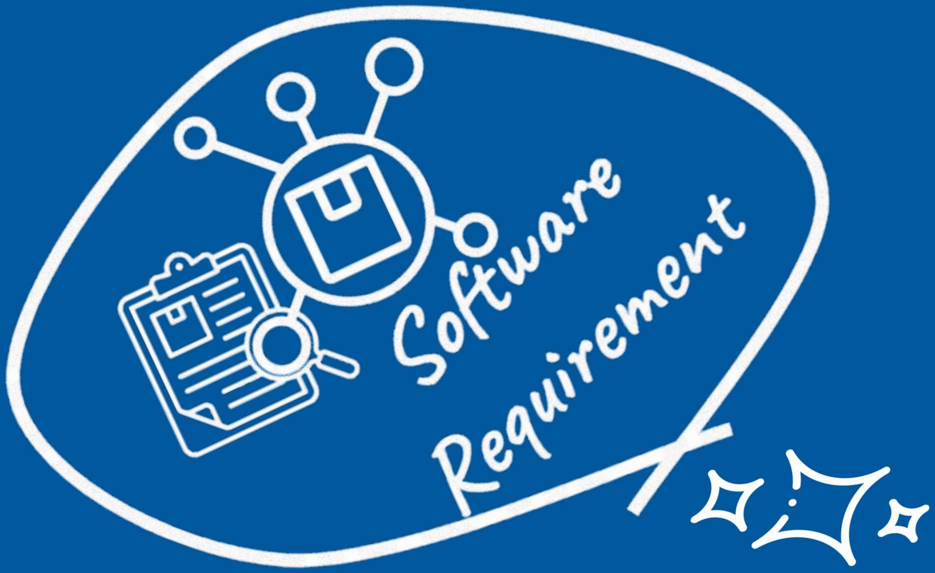


1305217

Software Requirements Analysis and Specification

Academic Year 2025/01

School of Information Technology, Mae Fah Luang University

AJ. Prasara Jakkaew

Email: prasara.jak@mfu.ac.th

Course Description

- Requirements elicitation;
- Requirements analysis;
- Requirements negotiation;
- Requirements specification;
- Requirements validation;
- Formal specification.
- Techniques and tools used to define;
- Document and ensure customer satisfaction

Course Objective

- To perform requirements engineering in the context of the most common software development life cycles and processes
- To introduce techniques for requirements elicitation and analysis
- To develop functional and non-functional requirements
- To create a requirements specification to communicate requirements to a broad set of stakeholders
- To effectively analyze requirements and prioritize accordingly

Google Class Code

For announcement

tnaxwix7

https://classroom.google.com/c/NzgzMTg5NDk0MDc3?cjc=tnaxwix7

Google Form :

For Quiz and Participant

https://forms.gle/ZKymd4teuaijDDvE7

Course Outline

Introduction to software requirement engineering

What is software requirement engineering, software requirement principles, software process requirements, functional and nonfunctional requirements, system requirements and software requirements

Requirements Process

- Process Models
- Process Actors
- Process Quality and Improvement
- Software Requirement Modeling (UML) for example; Use case diagram, Activity Diagrams, Workflow

Requirements Elicitation

- Requirements Sources
- Elicitation Techniques

Requirements Analysis

- Requirements Classification
- Conceptual Modeling
- Architectural Design and Requirements Allocation
- Requirements Negotiation
- Formal Analysis

Course Outline

Requirements Specification

- System Definition Document
- System Requirements Specification
- Software Requirements Specification

Requirements Validation

- Requirements Reviews
- Prototyping
- Model Validation
- Acceptance Tests

Lesson Plan: Before Midterm Exam

Week\	Date	Topics
1	4 Aug 2025	Course Introduction
2	11 Aug 2025 (Holiday)	No Class
3	18 Aug 2025 (Sci Day)	Chapter 1: Introduction to software requirement engineering
4	25 Aug 2025	Chapter 2: Requirements Engineering Process
5	1 Sep 2025	Chapter 3: Requirement Elicitation Part#1
6	8 Sep 2025	Assignment : Interview your stakeholders
7	15 Sep 2025	Assignment : Interview your stakeholders
8	22 Sep 2025 (no class)	Presentation 1

Lesson Plan: After Midterm Exam

Week	Date	Topics
9	29 Sep 2025 (Midterm)	No Class
10	6 Oct 2025	Chapter 4: Requirement Analysis
11	13 Oct 2025 (holiday)	No Class (Workshop Checklist & Consult) ONLINE
12	20 Oct 2025	Chapter 5: Requirement Specification
13	27 Oct 2025	Chapter 6: Requirement Validation
14	45964	Workshop Checklist & Consult
15	10 Nov 2025	Final Presentation
16	17 Nov 2025	Final Presentation
17	Final Exam	No Class

Evaluation

Midterm	25%
Final	25%
Group Project	30%
Quiz and Assignment	15%
Participation	5%

Project and Assignment

- **Software Requirements** will collaborate with **Database** and **Web application development**

- Group Work
  - 5-6 members / 1 group

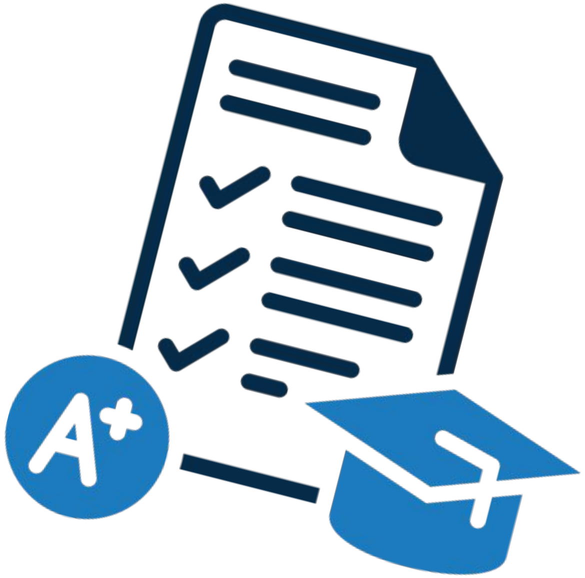
- Topic: Freestyle

- Link group member:

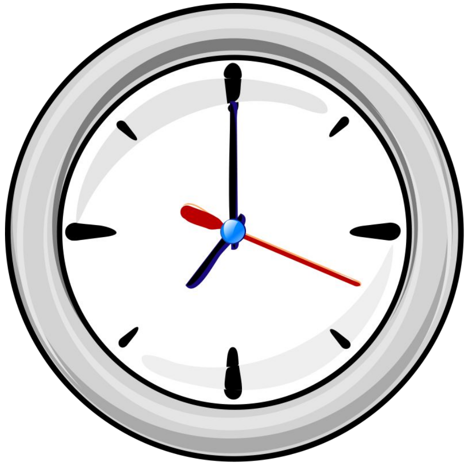
https://docs.google.com/spreadsheets/d/1_06SKZ_fFMz7KjypS0W14QSC_0liDPfo0C-iAYt4oEw/edit?usp=sharing

Grading

A	80-100
B+	75-79
B	70-74
C+	65-69
C	60-64
D+	55-59
D	50-54
F	0-49



Teaching Rules



- Students are required to attend all classes.
- **Check-in** class and sometime has a quiz
- **The assignment** has to send it to be on time before the deadline. If you turn in late, it means you will get 0 point.

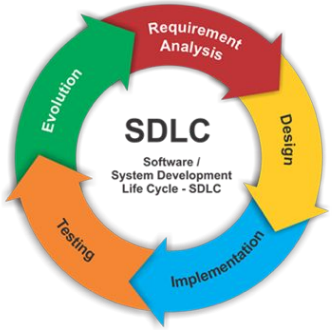
Introduction to Software Requirements

What is

Software **D**evelopment **L**ife **C**ycle

?

Software Development Life Cycle (SDLC) model is a simplified representation of a "Software Development Process Models".



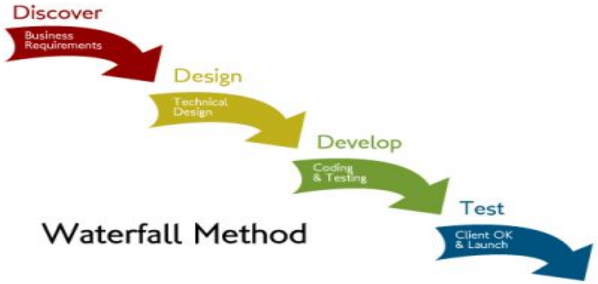
- SDLC is a framework defining tasks performed at each step in the software development process.

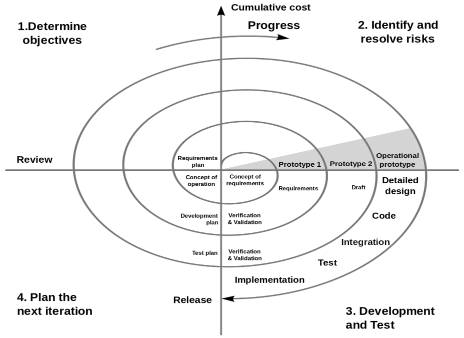
- The roadmap to building high quality software products is software process.

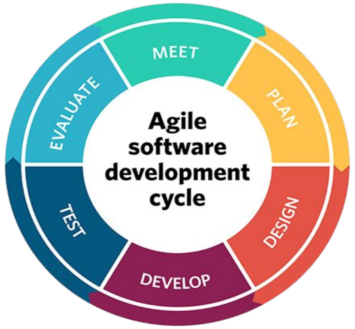
- Different life cycle models may plan the necessary development activities to phases in different ways.

- Following are the most important and **popular SDLC models** followed in the industry
  - Waterfall Model
  - V-Modell
  - Spiral Model
  - Iterative Mode

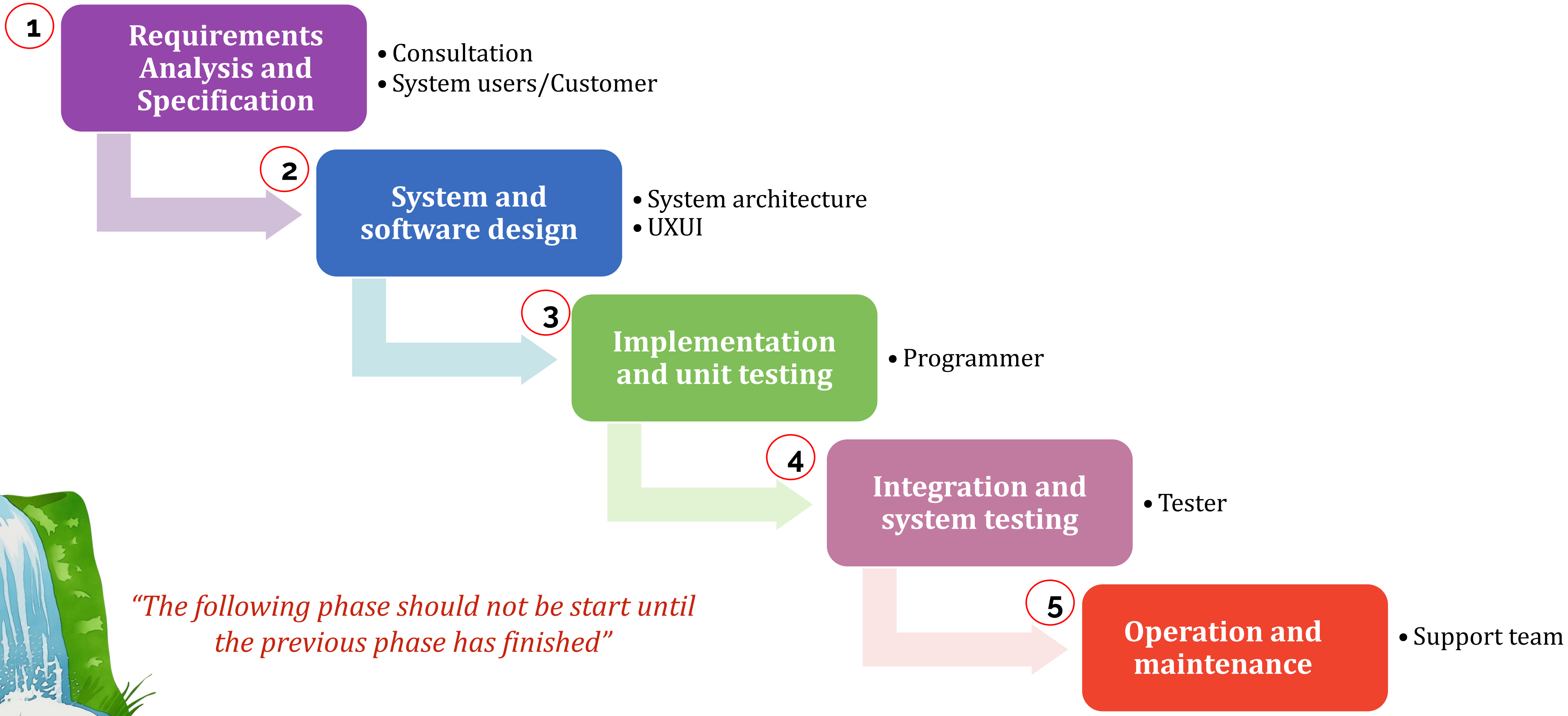
  Other related methodologies are Agile Model, Incremental Model, Rapid Application Development Model (RAD) Model, and Prototyping Models.







Waterfall Model



1 Requirements Analysis and Specification
- Consultation
- System users/Customer

2 System and software design
- System architecture
- UXUI

3 Implementation and unit testing
- Programmer

4 Integration and system testing
- Tester

5 Operation and maintenance
- Support team

“The following phase should not be start until the previous phase has finished”

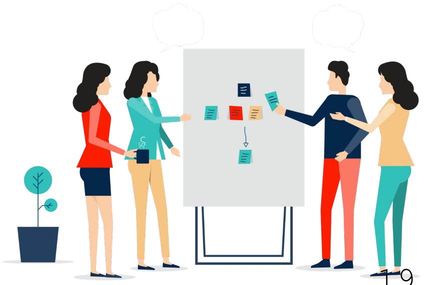
Waterfall Model

1 Requirements Analysis and Specification

The aim of the requirement analysis and specification phase is to understand the exact requirements of the customer and document them properly.

This phase consists of two different activities.

1. **Requirement gathering and analysis:** Firstly all the requirements regarding the software are gathered from the customer and then the gathered requirements are analyzed.
2. **Requirement specification:** These analyzed requirements are documented in a software requirement specification (SRS) document. SRS document serves as a contract between development team and customers.



https://www.geeksforgeeks.org/software-engineering-classical-waterfall-model/

Why do we need user research?

Customer insights

Understand what Problems customers want solved

Surveys Interviews Focus Group

Function Priority

Analyze how important These problems are to customers

Usability

What products and features we can develop to solve these problems

Resource from Slideshare :: Joan Lumanauw

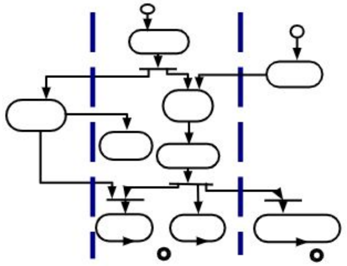
Waterfall Model

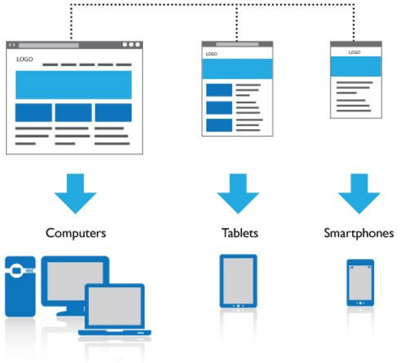
2 System and software design

The aim of the design phase is to transform the requirements specified in the SRS document into a structure that is suitable for implementation in some programming language. It defines the overall software architecture together with high level and detailed design. All this work is documented as a Software Design Document (SDD).

Example:

- Plan the programming language, for Example Java, PHP, ASP.net
- or database like Oracle, MySQL, etc.
- Or other high-level technical details of the project





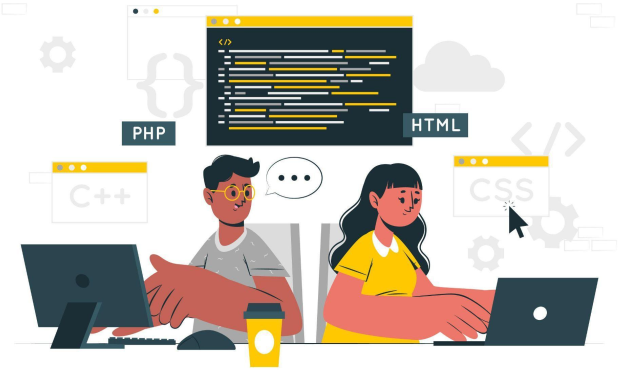
https://www.geeksforgeeks.org/software-engineering-classical-waterfall-model/

Waterfall Model

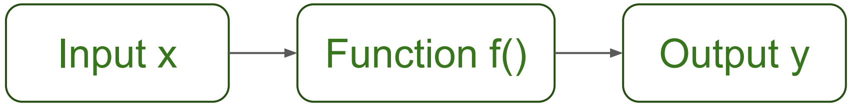
3 Implementation and unit testing

This phase aims to transform the requirements gathered in the SRS into a suitable form which permits further coding in a programming language.

With inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.



Unit Testing



Waterfall Model

4 Integration and system testing

All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.

System testing consists three different kinds of testing activities as described below :

- **Alpha testing:** Alpha testing is the system testing performed by the development team.
- **Beta testing:** Beta testing is the system testing performed by a friendly set of customers.
- **Acceptance testing:** After the software has been delivered, the customer performed the acceptance testing to determine whether to accept the delivered software or to reject it.

https://www.geeksforgeeks.org/software-engineering-classical-waterfall-mod

Waterfall Model

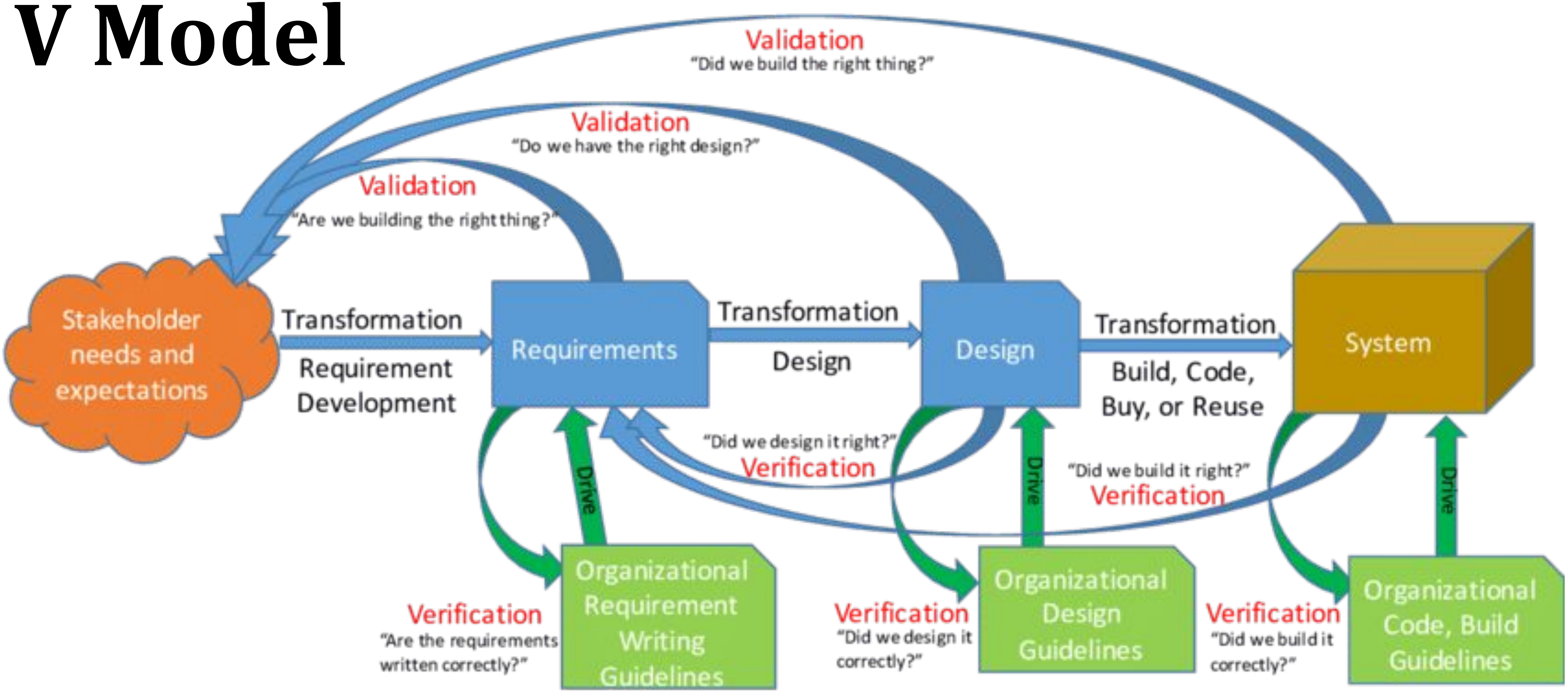
5 Operation and maintenance

Maintenance is the most important phase of a software life cycle. The effort spent on maintenance is the 60% of the total effort spent to develop a full software.

There are some issues which come up in the client environment. To fix those issues, patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

https://www.geeksforgeeks.org/software-engineering-classical-waterfall-model/

V Model

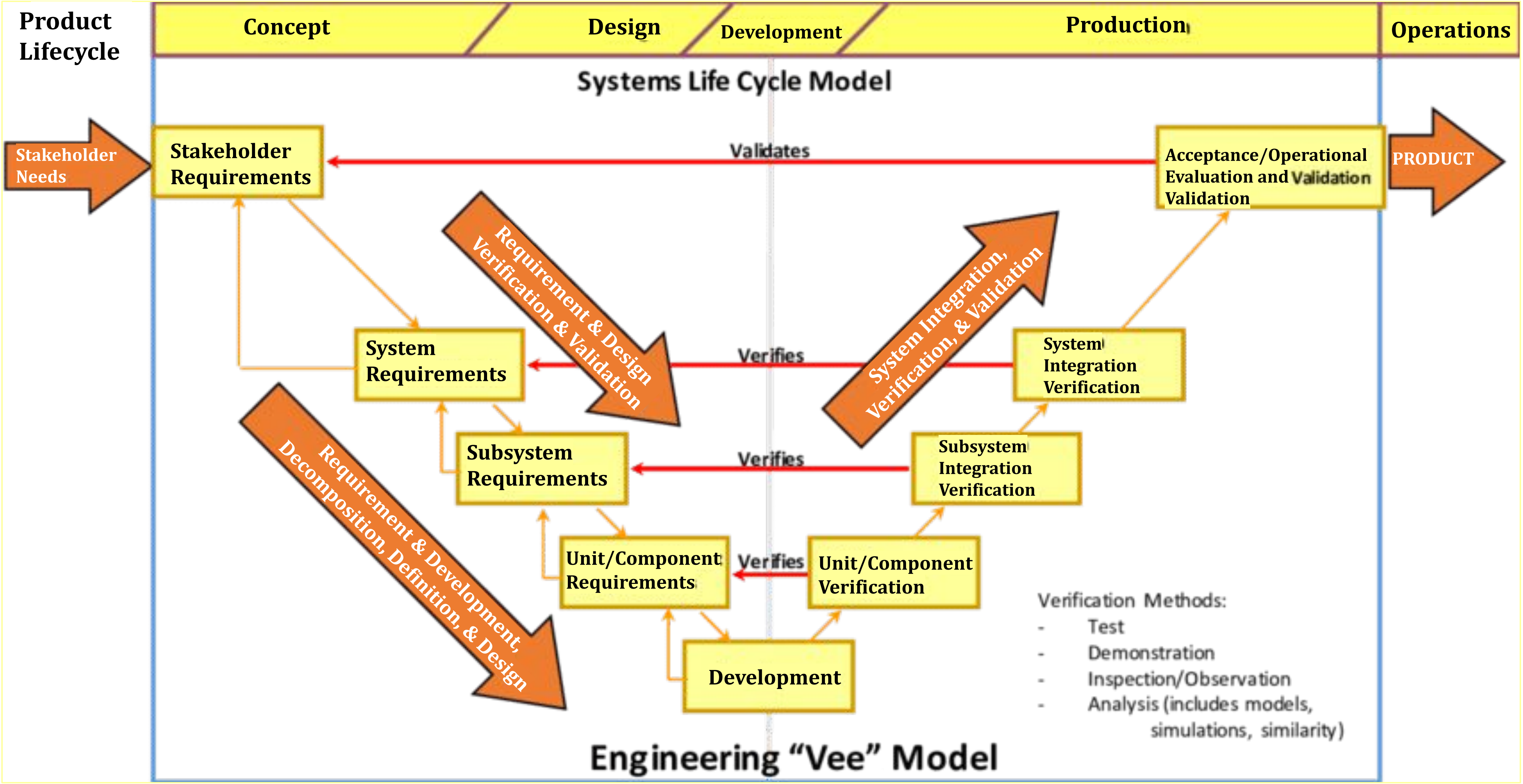

Validation
"Did we build the right thing?"
Validation
"Do we have the right design?"
Validation
"Are we building the right thing?"
Stakeholder needs and expectations
Transformation
Requirement Development
Requirements
Transformation
Design
Design
Transformation
Build, Code, Buy, or Reuse
System
"Did we design it right?"
Verification
"Did we build it right?"
Verification
Drive
Drive
Drive
Verification
"Are the requirements written correctly?"
Organizational Requirement Writing Guidelines
Verification
"Did we design it correctly?"
Organizational Design Guidelines
Verification
"Did we build it correctly?"
Organizational Code, Build Guidelines

Verification and validation are the processes of confirming that systems engineering artifacts generated during the transformation processes are acceptable

*Journal Reference by **Mike Ryan** and **Louis S Wheatcraft** On the Use of the Terms Verification and Validation : July 2017INCOSE International Symposium 27(1):1277-1290*

V Model

Verification and Validation and the Systems Engineering " V " Model

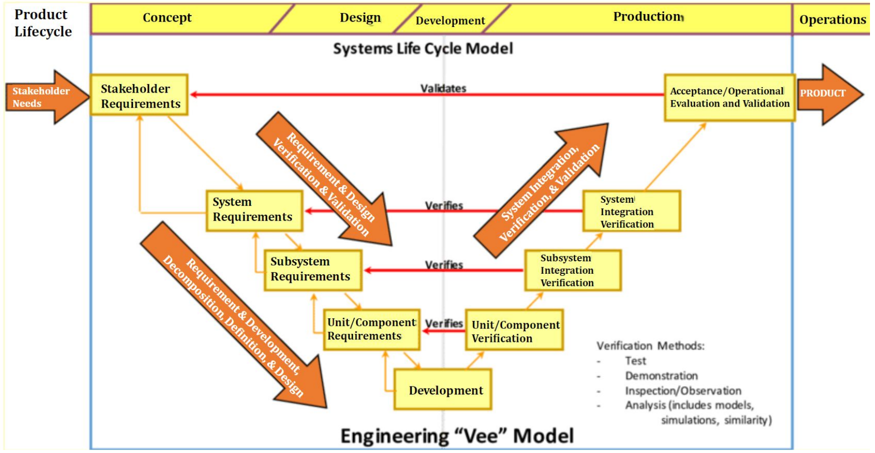

Product Lifecycle
Concept
Design
Development
Production
Operations
Systems Life Cycle Model
Stakeholder Needs
Stakeholder Requirements
Validates
Acceptance/Operational Evaluation and Validation Validation
PRODUCT
Requirement & Design Verification & Validation
System Integration, Verification, & Validation
System Requirements
Verifies
System Integration Verification
Subsystem Requirements
Verifies
Subsystem Integration Verification
Requirement & Development Decomposition, Definition, & Design
Unit/Component Requirements
Verifies
Unit/Component Verification
Development
Verification Methods:
- Test
- Demonstration
- Inspection/Observation
- Analysis (includes models, simulations, similarity)
Engineering "Vee" Model

Journal Reference by ***Mike Ryan*** *and* ***Louis S Wheatcraft*** *On the Use of the Terms Verification and Validation : July 2017INCOSE International Symposium 27(1):1277-1290*

Requirement Engineering

What is requirements engineering?

What is requirements engineering?


Product Lifecycle
Concept
Design
Development
Production
Operations
Systems Life Cycle Model
Stakeholder Needs
Stakeholder Requirements
Validates
Acceptance/Operational Evaluation and Validation
PRODUCT
Requirement & Design Verification & Validation
System Integration, Verification, & Validation
System Requirements
Verifies
System Integration Verification
Subsystem Requirements
Verifies
Subsystem Integration Verification
Requirement & Development, Decomposition, Definition, & Design
Unit/Component Requirements
Verifies
Unit/Component Verification
Development
Verification Methods:
- Test
- Demonstration
- Inspection/Observation
- Analysis (includes models, simulations, similarity)
Engineering "Vee" Model

- Requirement engineering is usually the **first step** in any software development life cycle

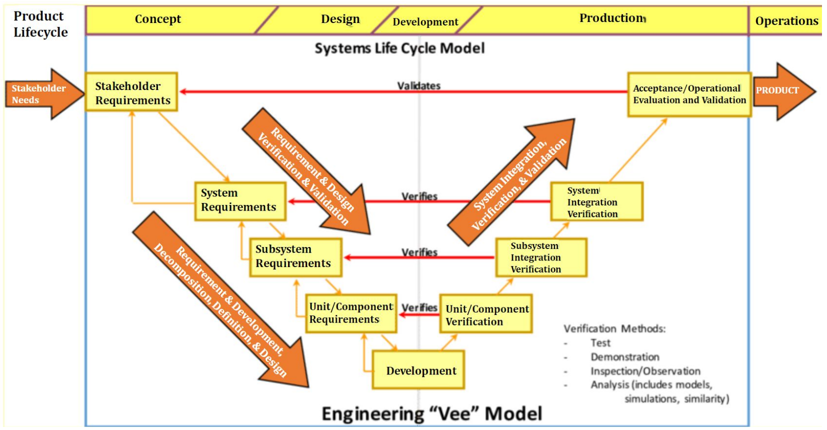
- The process to **gather** the software **requirements** from client, analyze and document them is known as requirement engineering.

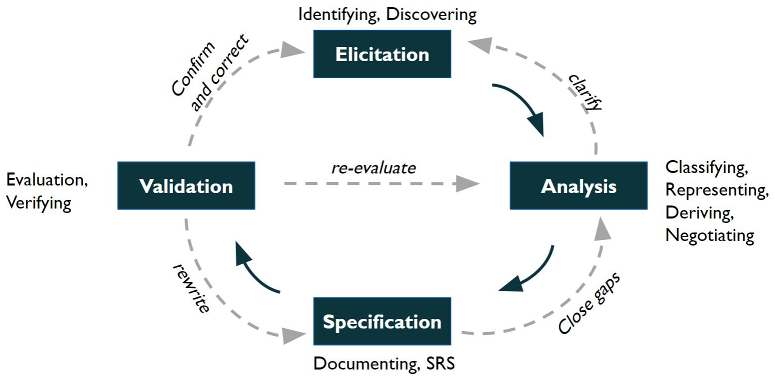
- The **goal** of requirement engineering is to create and maintain detailed **'System Requirements Specification' documents.**

- **Ensures** your **software** will **meet the user expectations,** and ending up with a high-quality software

Resource by https://medium.com/omarelgabrys-blog/software-engineering-process-activities-part-3-ca1ef6818fd6

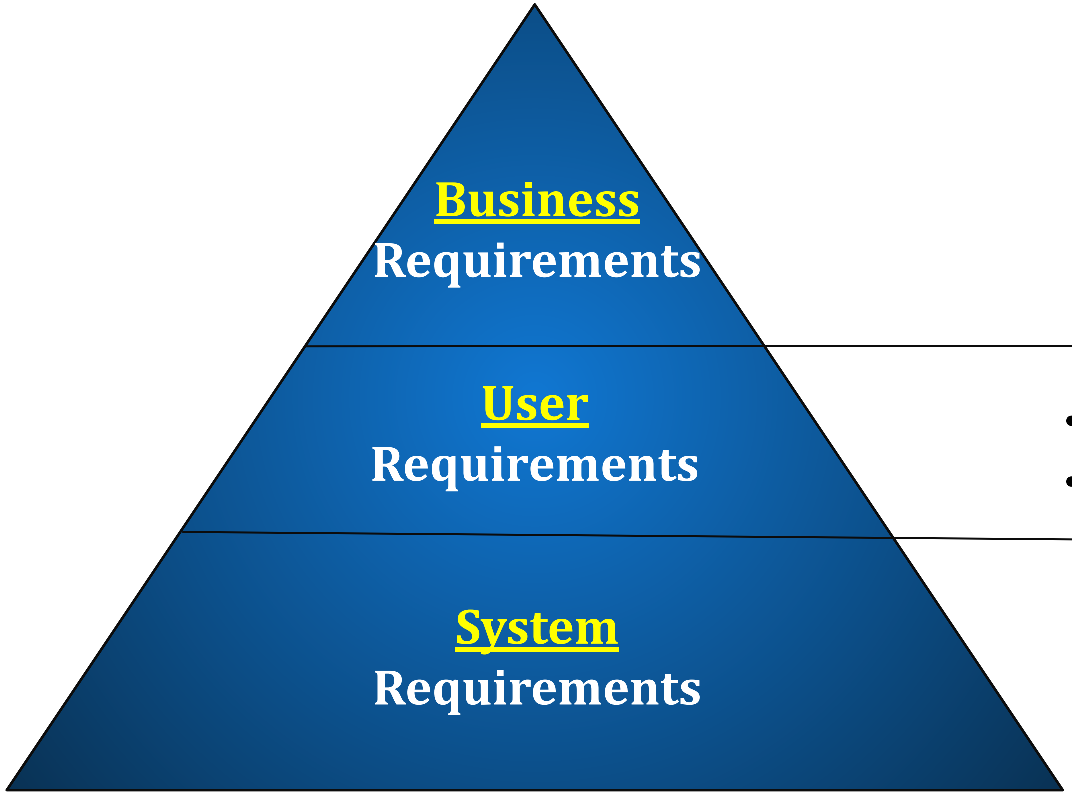
What is requirements engineering?


Product Lifecycle
Concept
Design
Development
Production
Operations
Systems Life Cycle Model
Stakeholder Needs
Stakeholder Requirements
Validates
Acceptance/Operational Evaluation and Validation
PRODUCT
Requirement & Design Verification & Validation
System Integration, Verification, & Validation
System Requirements
Verifies
System Integration Verification
Requirement & Development, Decomposition, Definition, & Design
Subsystem Requirements
Verifies
Subsystem Integration Verification
Unit/Component Requirements
Verifies
Unit/Component Verification
Development
Verification Methods:
- Test
- Demonstration
- Inspection/Observation
- Analysis (includes models, simulations, similarity)
Engineering "Vee" Model



- Software requirements engineering is the **process of determining** what will be developed in a software system.
- The **<u>4 steps</u>** of software requirements engineering are
  - Requirements **elicitation**,
  - Requirements **analysis**,
  - Requirements **specification**,
  - Requirements **validation**.

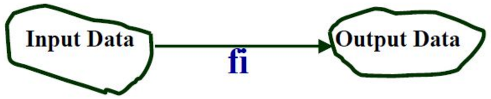
Understand Requirements Types



- Stakeholders point of view
- Project scope
- Business objectives

- Users point of view (UI)
- User Goals

- Functional Requirements
- What the system does
- Non-Functional Requirements



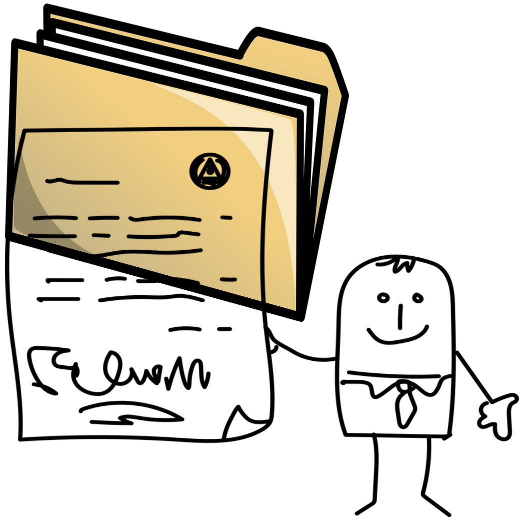
What is Requirements Engineering?

Agreement

Understanding

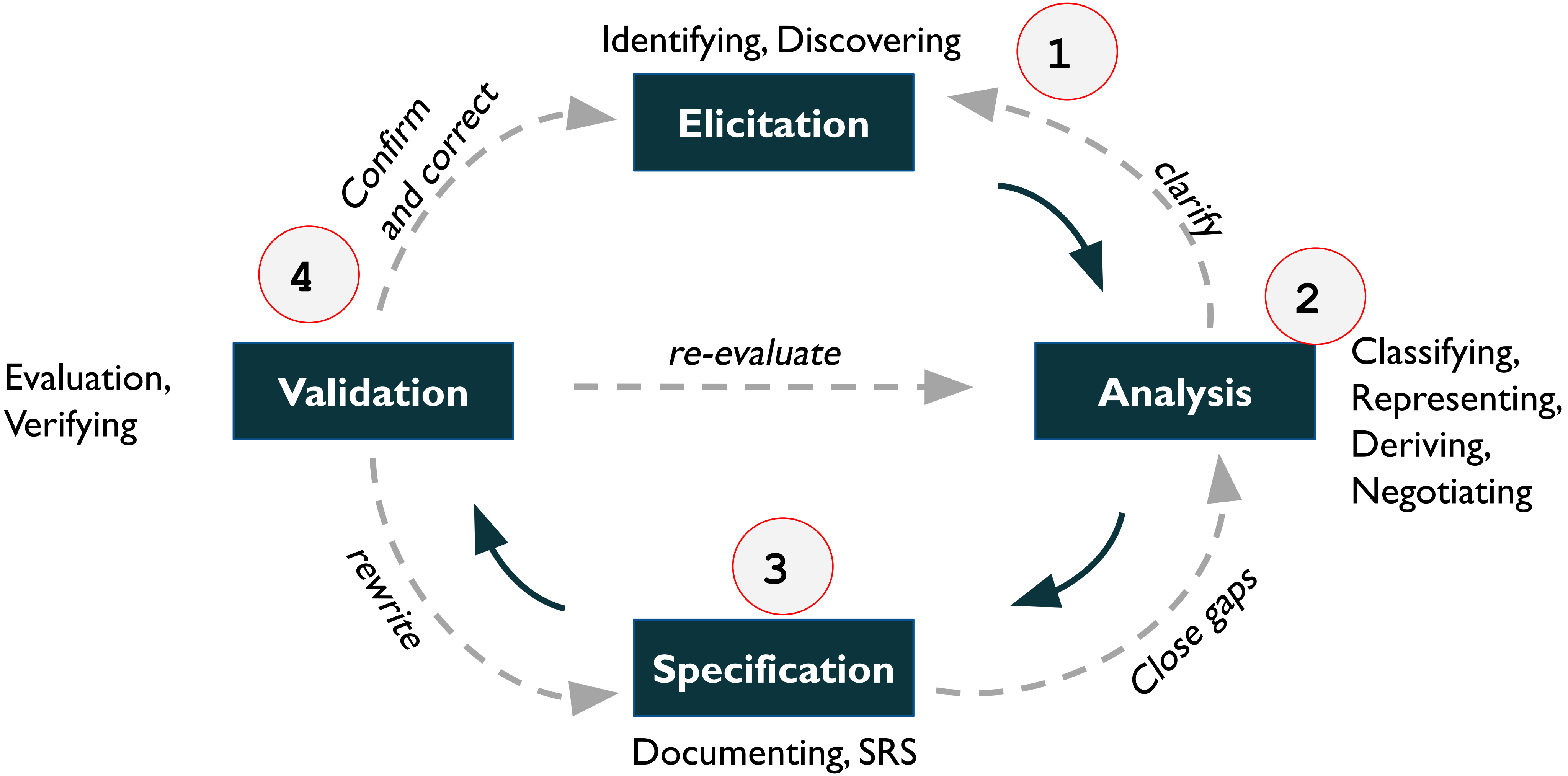
+

Systematic



Documenting

The Requirement Development Processes



Resource by D. Monett-Europe Week2015, University of Hertfordshire, Hatfield

Requirement Engineering

We will learn……

Software Requirements Fundamentals

Software Process Model

Requirement Elicitation

Requirement Analysis

Requirement Validation

Practical Consideration

Requirement Documentation