Introduction to Software Engineering

Day 1

Anjali Kulkarni

May 2024

Software Engineering

Software engineering is a branch of computer science – aims towards building a quality product

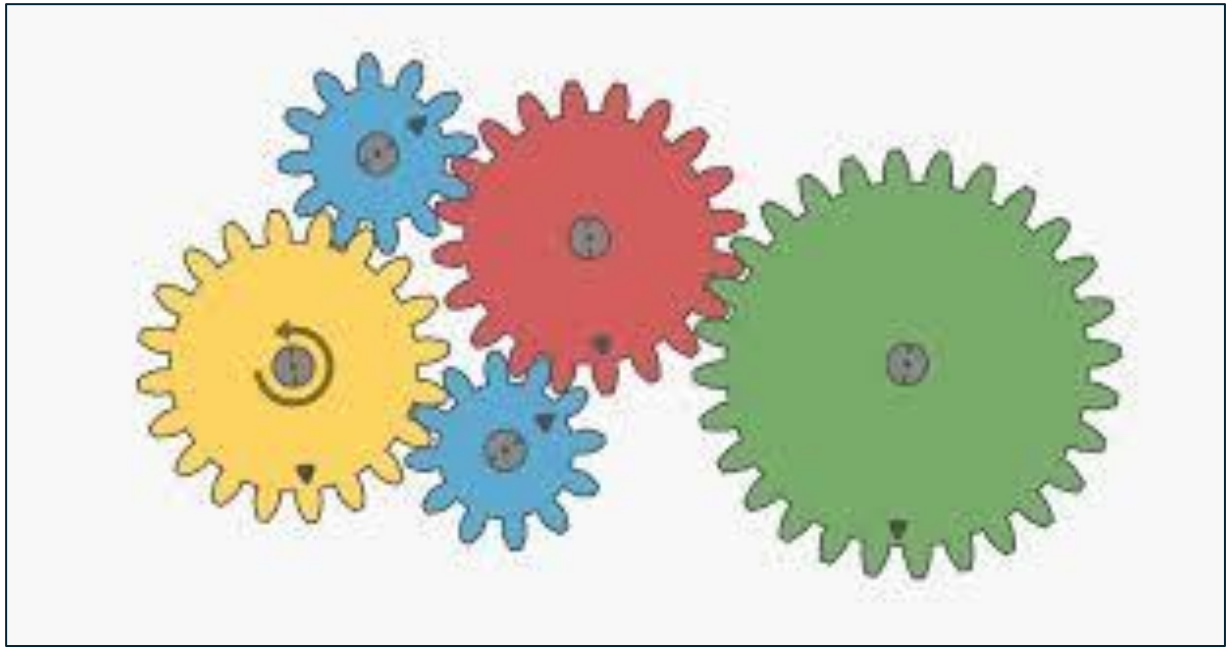
Building blocks

- Software
- Software Product
- Software Model
- Software Process

What is a software?

<u>Instructions that Power Your Devices</u>

- System software
  - OS, processors
- Application software
  - User-oriented tasks

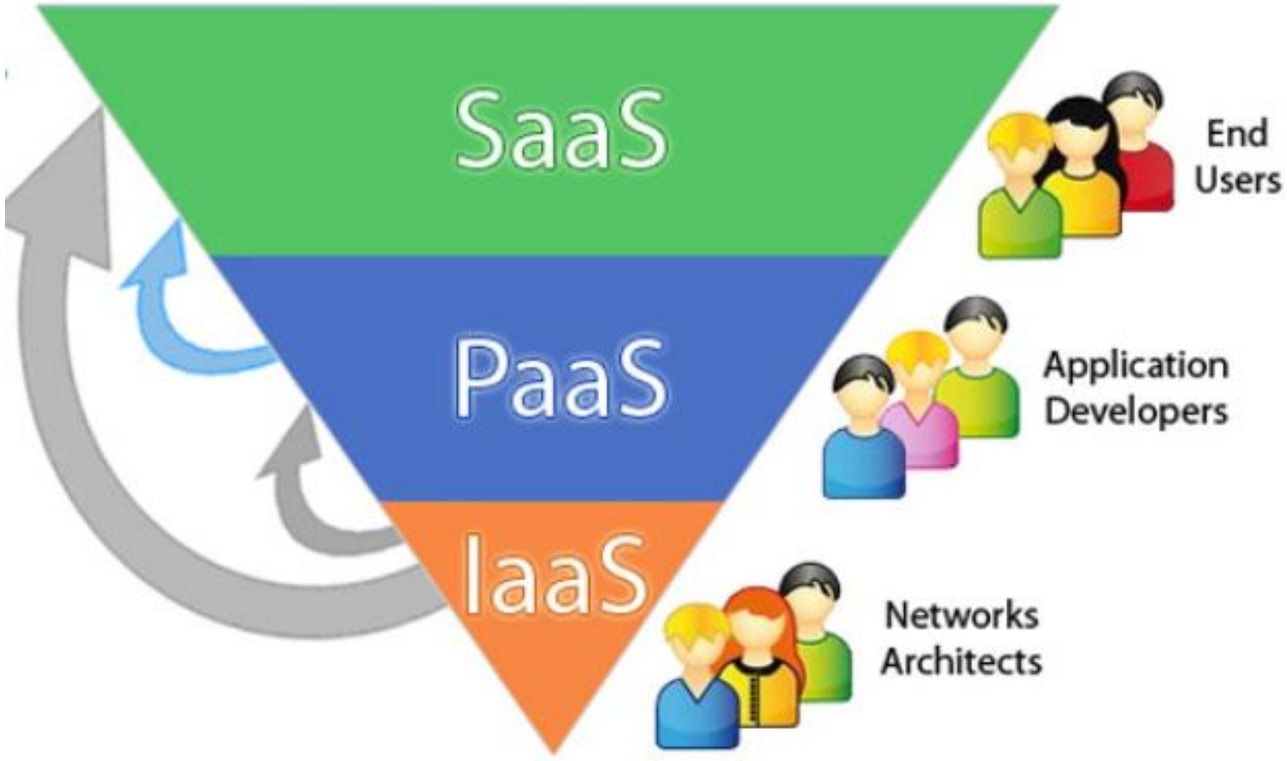


What is a Software Product?

<u>Delivering Value Through Software</u>

- Increased Efficiency
- Improved Productivity
- Streamlined Communication
- Enhanced Entertainment

Types of Software Product



- Netflix, Instagram, Zomato, Google Workspace

- Canva, Github, AWS Lambda

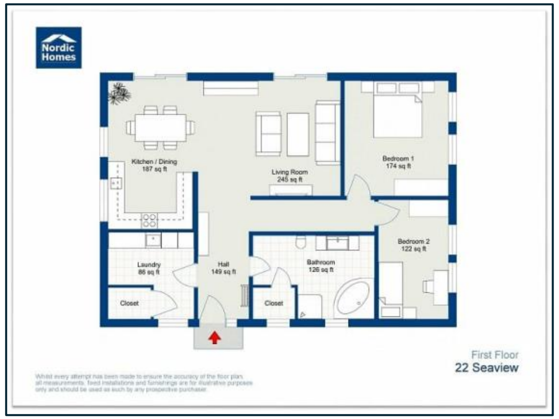
- GCP, AWS, Azure

What is Software Model?

<u>Visual or textual representation of a software system</u>

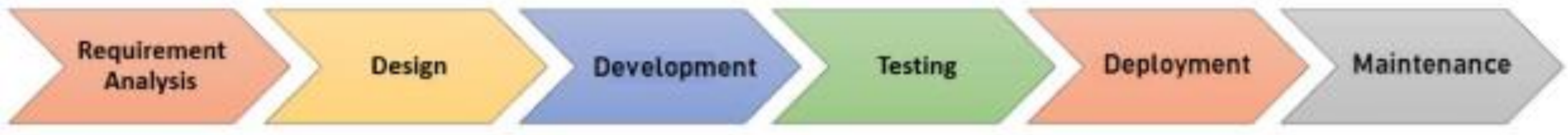
It helps understand

- Structure
- Behaviour
- Functionality


Nordic Homes
Kitchen / Dining
187 sq ft
Living Room
245 sq ft
Bedroom 1
174 sq ft
Bedroom 2
122 sq ft
Laundry
86 sq ft
Hall
149 sq ft
Bathroom
126 sq ft
Closet
Closet
First Floor
22 Seaview
Whilst every attempt has been made to ensure the accuracy of the floor plan, all measurements, fixed installations and furnishings are for illustrative purposes only and should be used as such by any prospective purchaser.

What is Software Process?

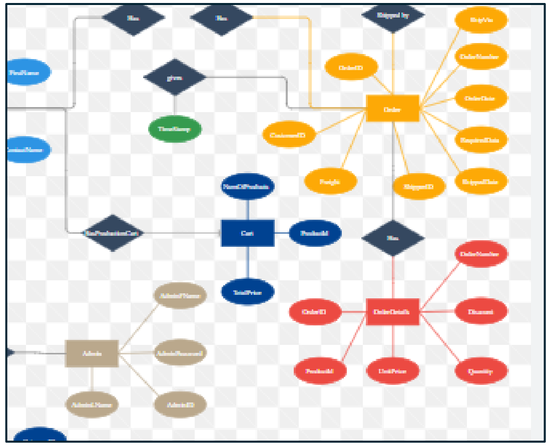
Structured set of activities required to develop a software system

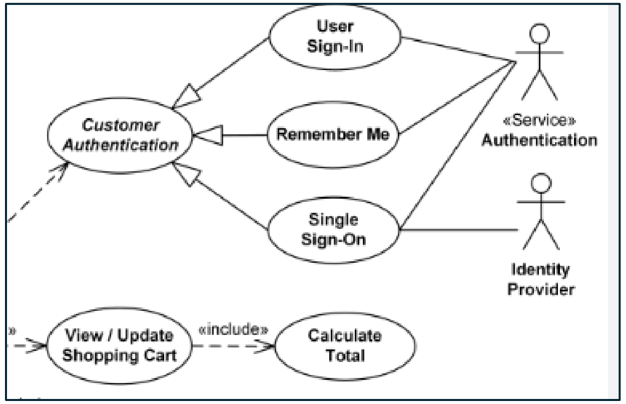


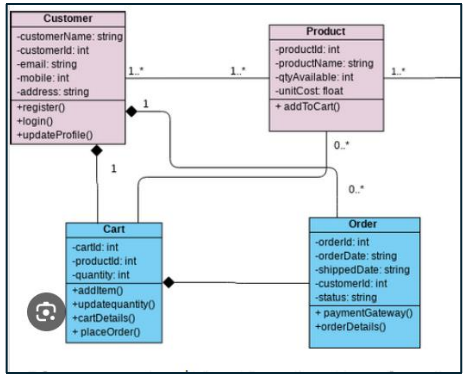
Efficiently ***Effectively*** ***On time***

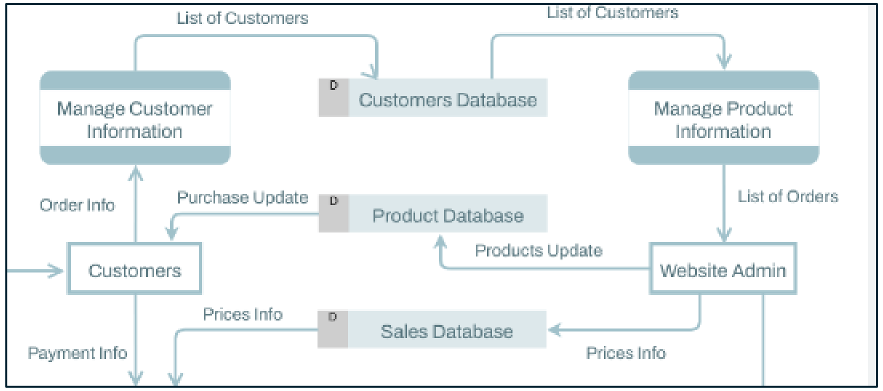
Types of Software Models

- ER Diagrams
- Use case Diagrams
- DFDs
- Class Diagrams

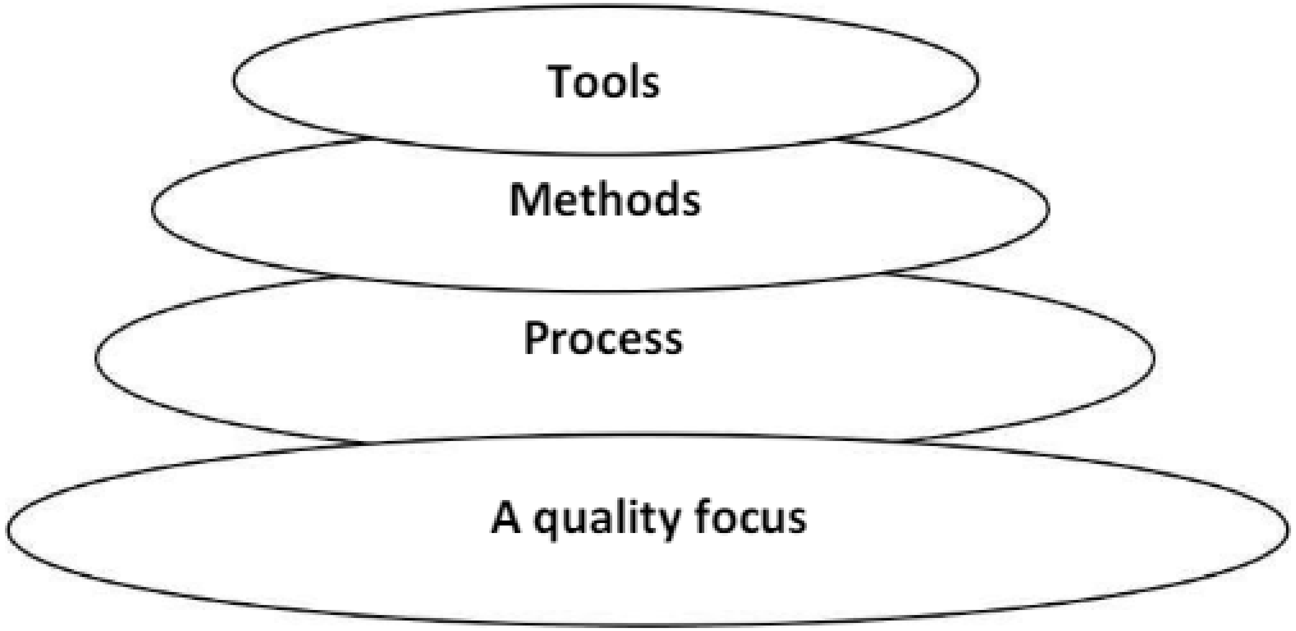






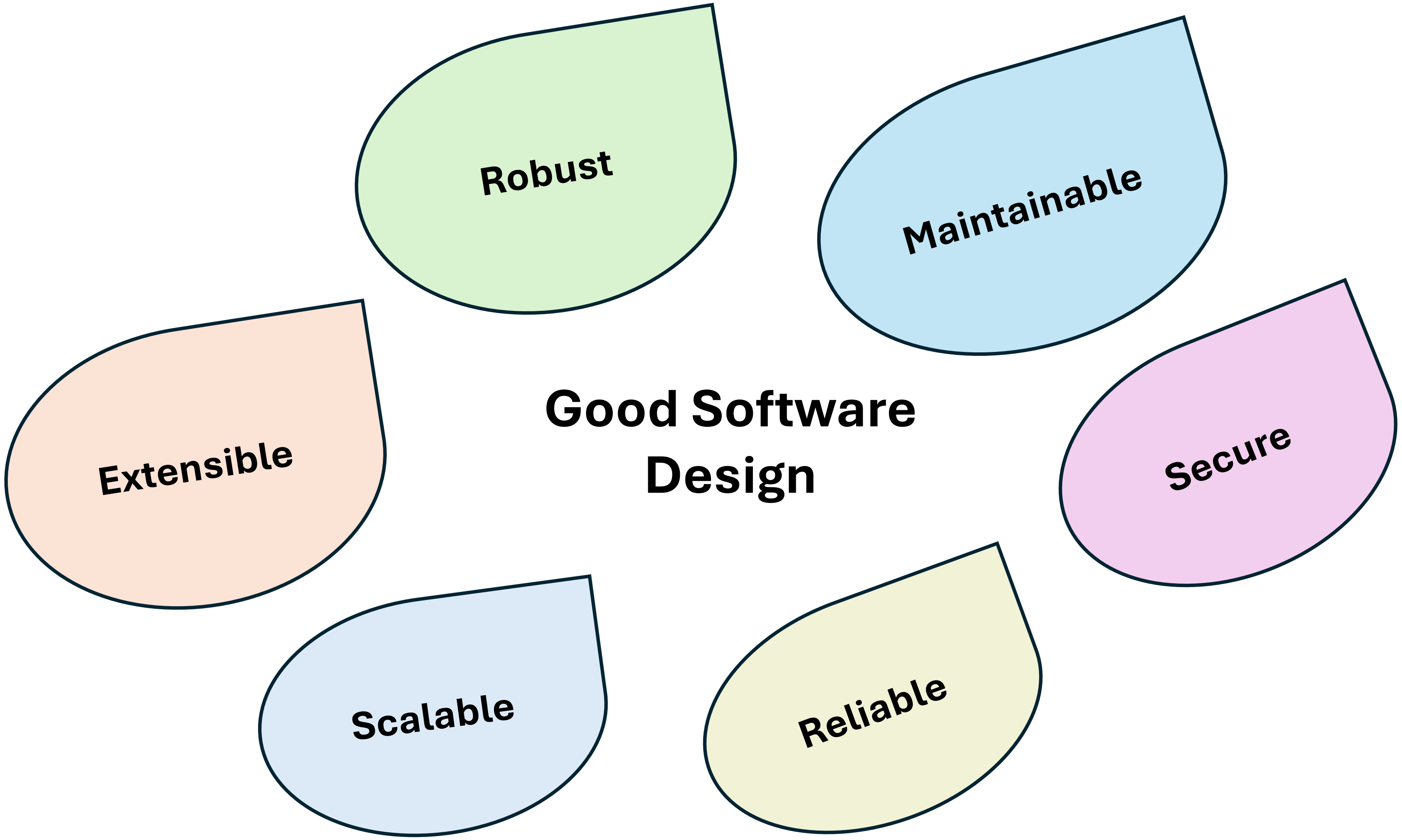


SOFTWARE ENGINEERING - A LAYERED TECHNOLOGY



Software Engineering Layers

Software Engineering - Attributes

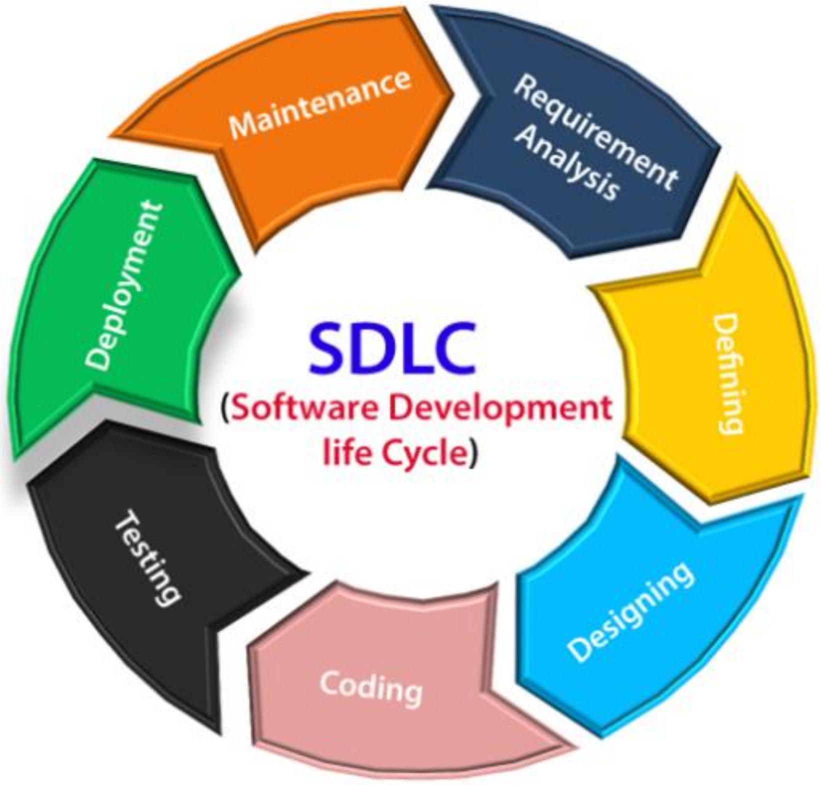


Software Engineering - Importance

- Improved Quality
- Increased Productivity
- Reduced Costs
- Better Communication
- Fosters Confidence

Software Development Life Cycles - SDLC

Roadmap to Building High-Quality Software



Waterfall Model

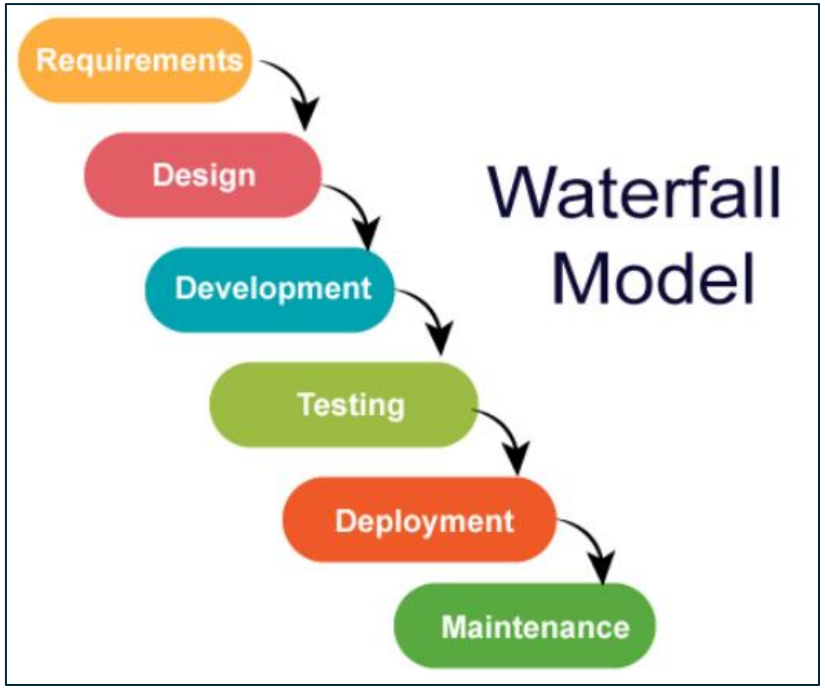
<u>Sequential approach</u>

<u>When to use</u>

- For very small and simple projects
- Requirements are clear

<u>Downside</u>

- Inflexible



Agile Model

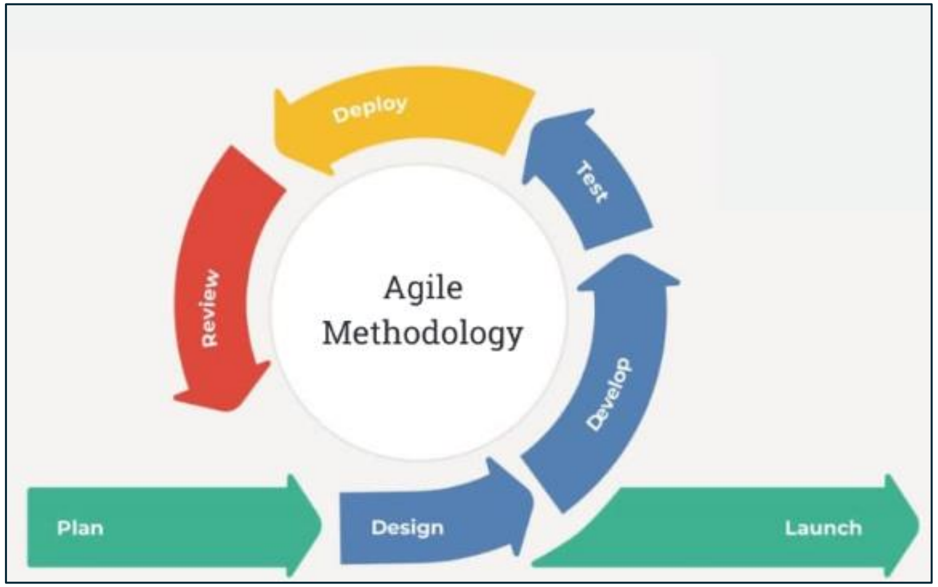
Iterative and incremental approach

When to use

- For Large complex projects

Downside

- Fragmented output



Spiral Model

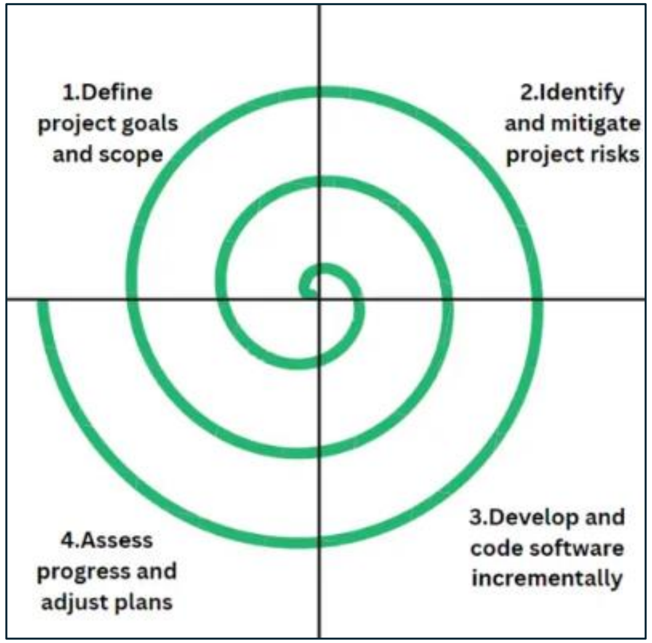
Risk-driven approach

When to use

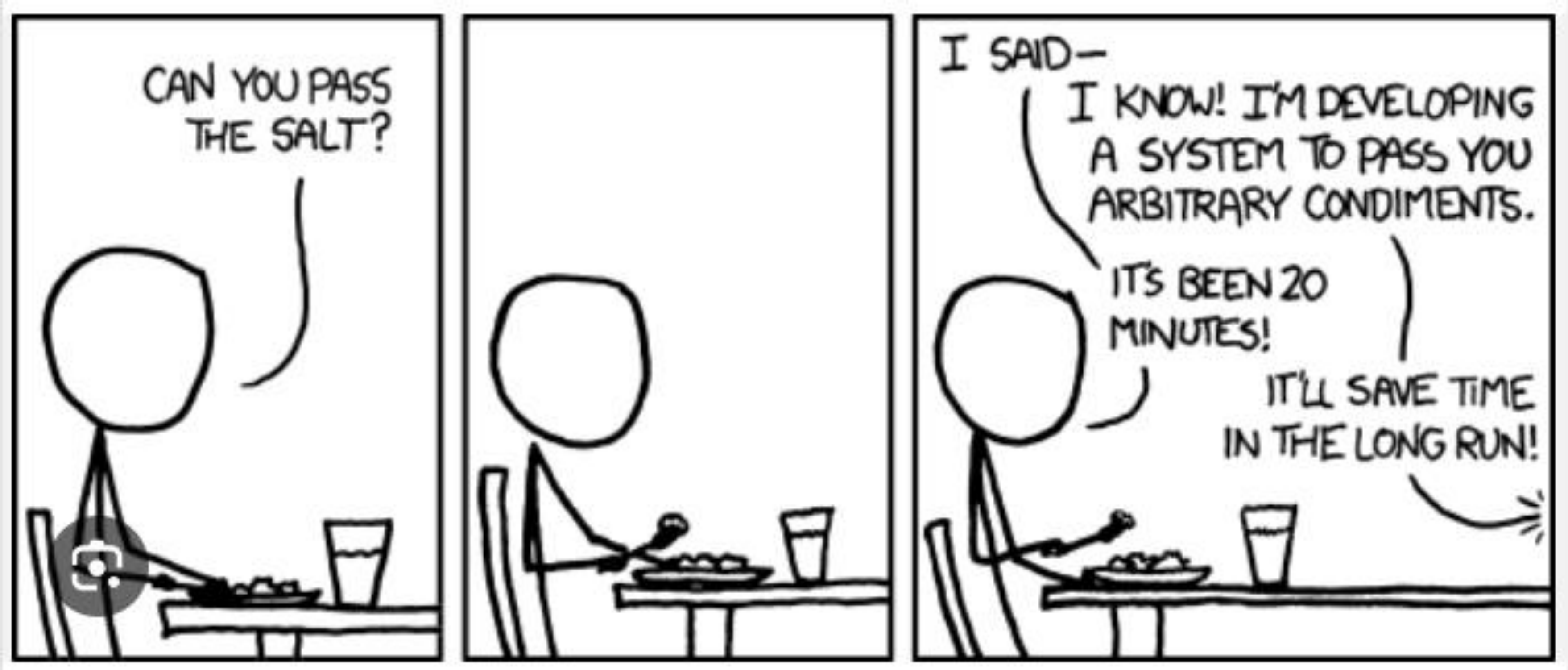
- For Large complex projects with unknowns

Downside

- Expensive
- Poor time management
- Needs excessive documentation


1.Define project goals and scope
2.Identify and mitigate project risks
3.Develop and code software incrementally
4.Assess progress and adjust plans

Choosing the right SDLC model

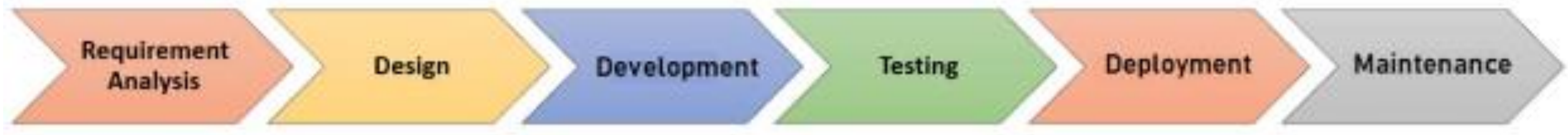

CAN YOU PASS THE SALT?
I SAID—
I KNOW! I'M DEVELOPING A SYSTEM TO PASS YOU ARBITRARY CONDIMENTS.
IT'S BEEN 20 MINUTES!
IT'LL SAVE TIME IN THE LONG RUN!

- Project size
- Complexity
- Req stability
- Timeline
- Budget
- Team expertise

Recap

- What are Building blocks of Software Engineering?
- What are types of Software Product?
- Name SDLC models
- List Stages in SDLC model
- What are attributes of a good software design?
- What is the end goal of Software Engineering?

Key Activities in the Software Process



Requirements Analysis: Understanding what the software system should do

Design: Defining the architecture, components, and interfaces of the system

Development: Writing the code for the system

Testing: Verifying that the system meets its requirements

Deployment: Releasing the system to its end users

Maintenance: Making changes to the system after it has been deployed

What are Requirements?

Description of the services provided by a system

Requirements Engineering

Process of defining services and constraints

- Who are the users of the system?

- What are the needs of the users?

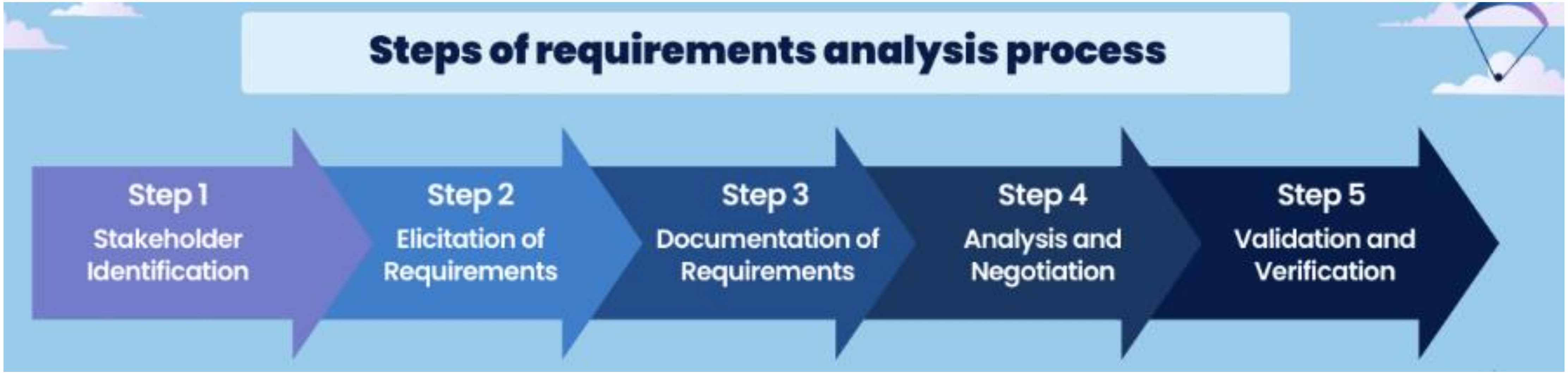
- What do we want the system to do?

- What does the system need to do to overcome end user challenges?

Requirements - Types

- **Functional** – Functionality that end user expects
- **Non-Functional** – Quality Attributes
- **Domain** – Rules/regulations that govern the domain

Requirements Engineering - Stages



Steps of requirements analysis process

Step 1 Stakeholder Identification

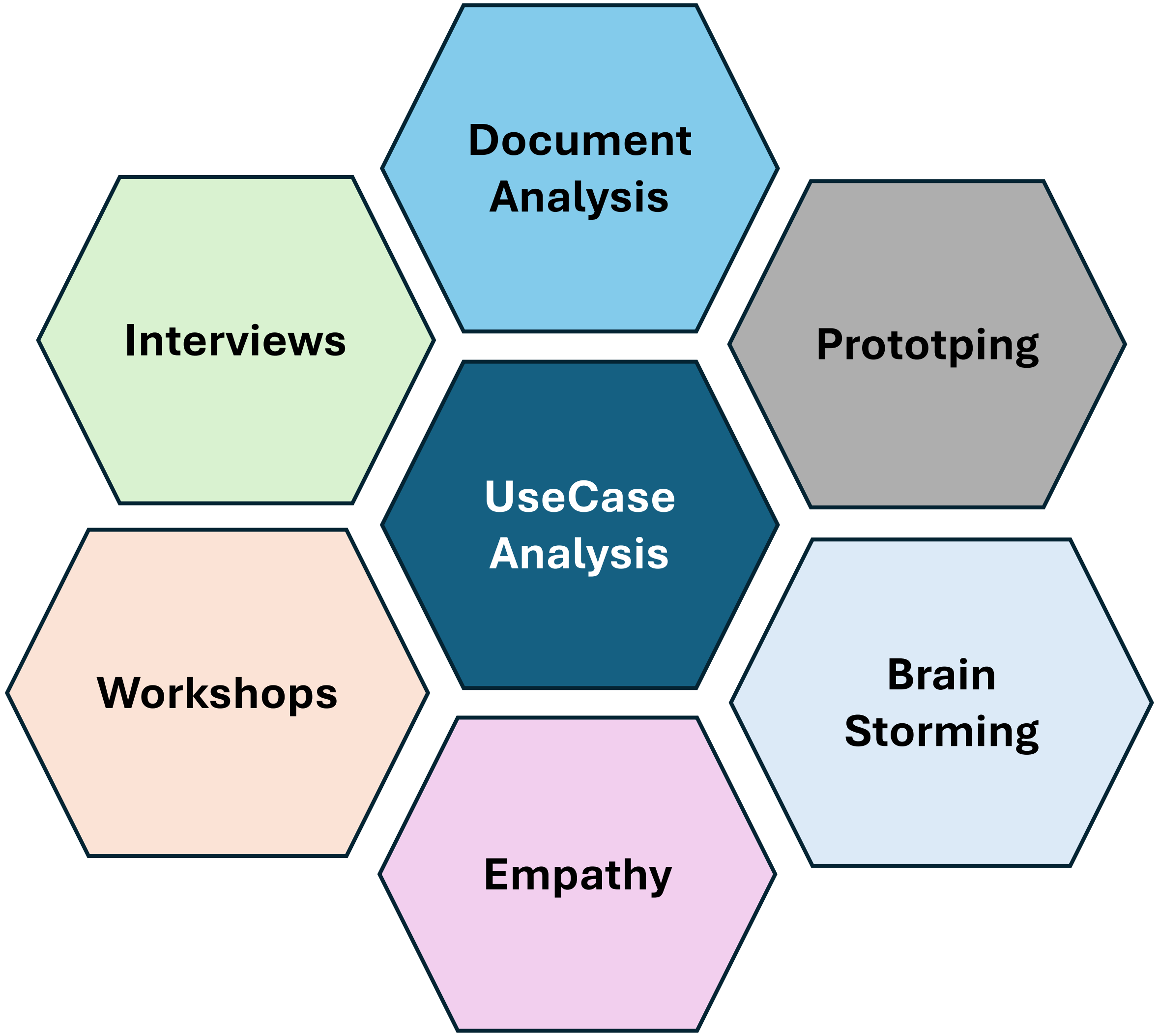
Step 2 Elicitation of Requirements

Step 3 Documentation of Requirements

Step 4 Analysis and Negotiation

Step 5 Validation and Verification

Requirements Elicitation - Techniques



Requirements Engineering - Importance

- Reduced rework
- Reduced delays
- Reduced risks
- Improved project estimation
- Increased product quality

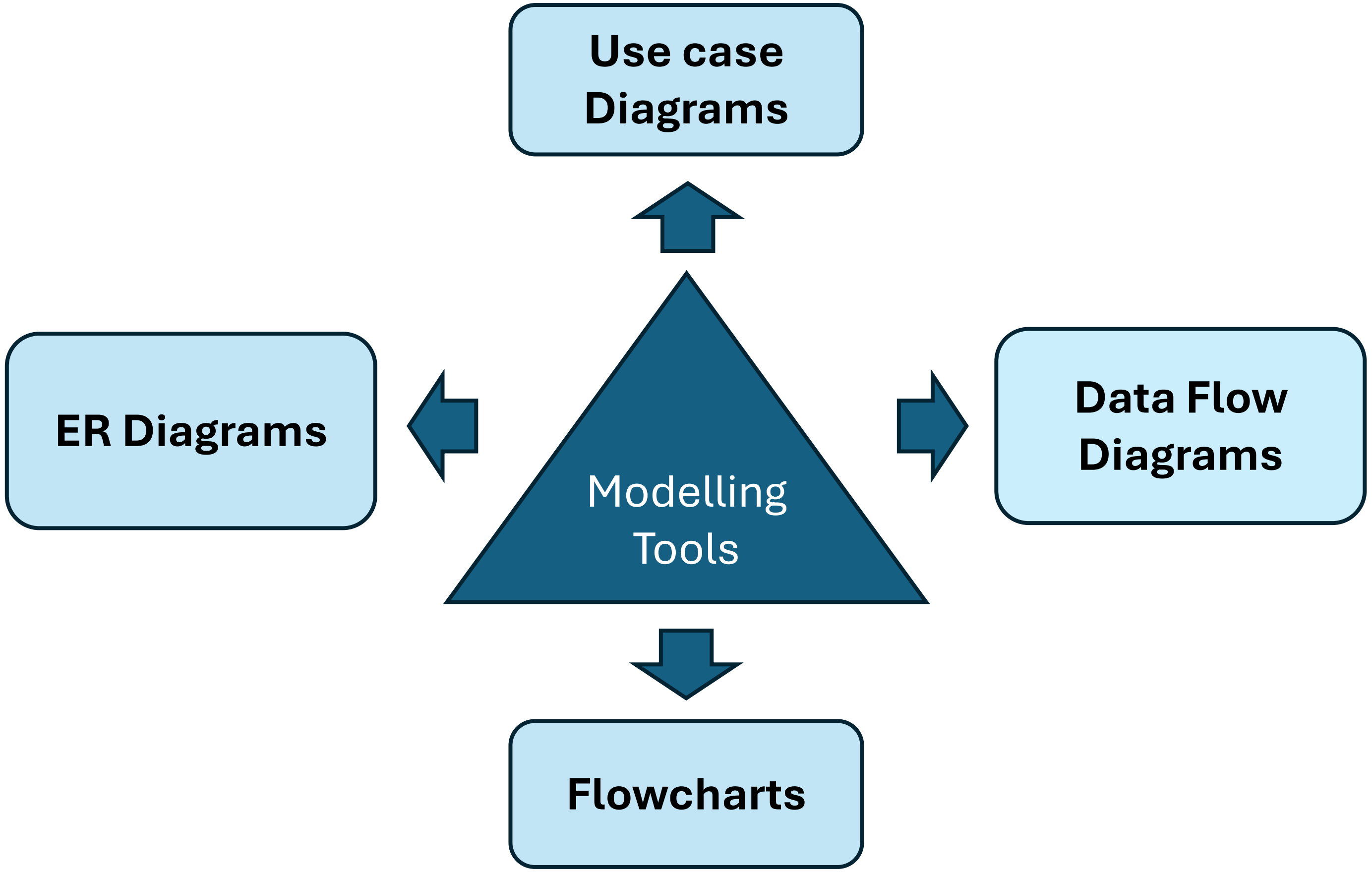
Requirement Analysis Modelling

Visual representation of requirements in textual format

- Depicting functional requirements

- Establish relationships between requirements

Requirements Modelling - Tools



Use case Diagrams

ER Diagrams

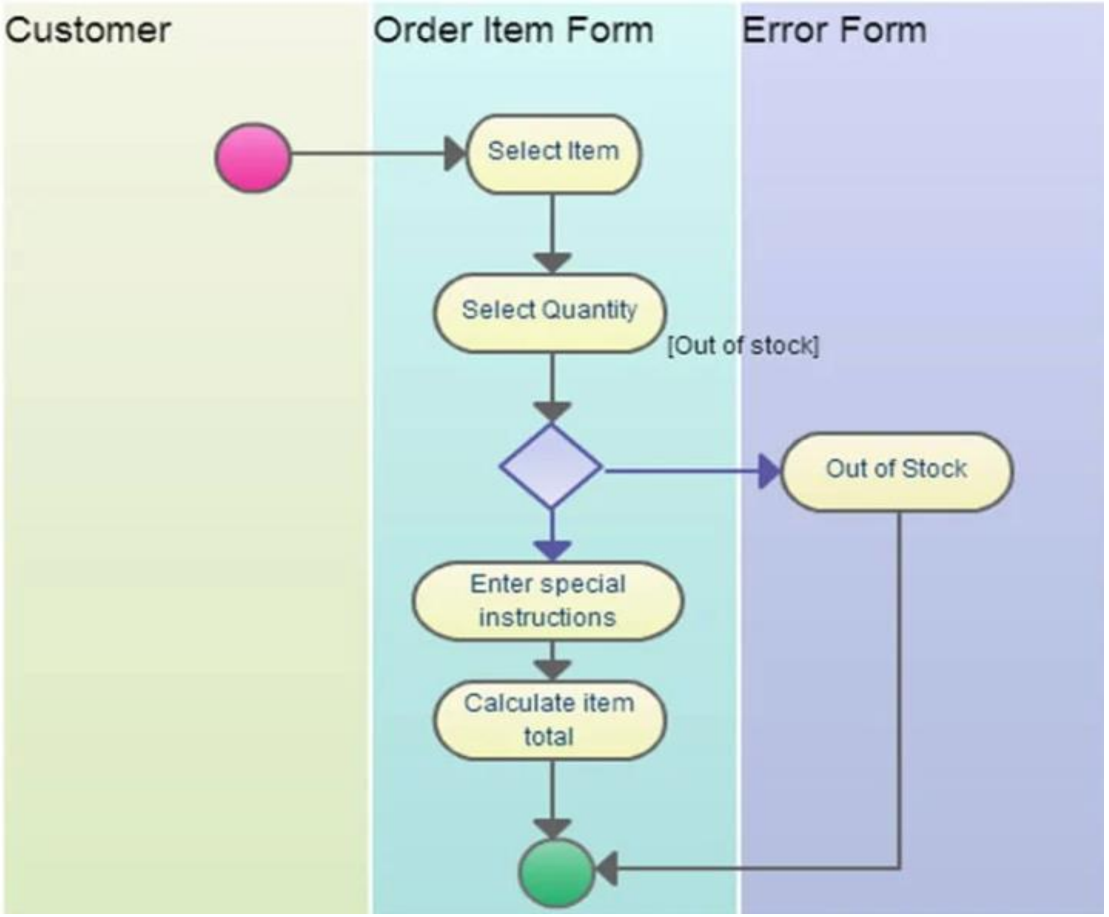
Modelling Tools

Data Flow Diagrams

Flowcharts

Flow Charts

Depicts Sequence of actions



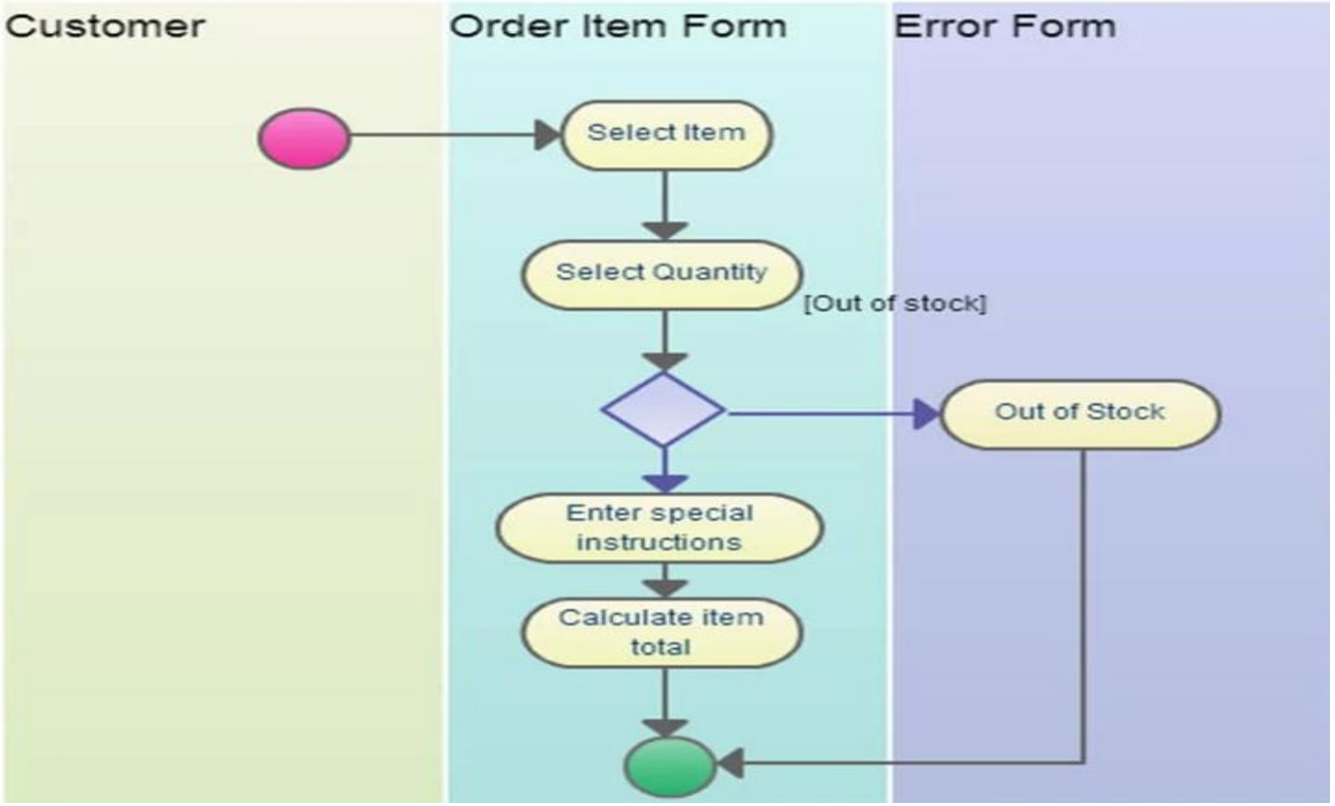
User stories

Describe how end-users interact with the system

- Requirements document – Textual format with many details
- User Stories – Usage perspective

User stories - Format

"As a [user role], I want to [goal] so that [benefit].“



Requirement Modelling - Value additions

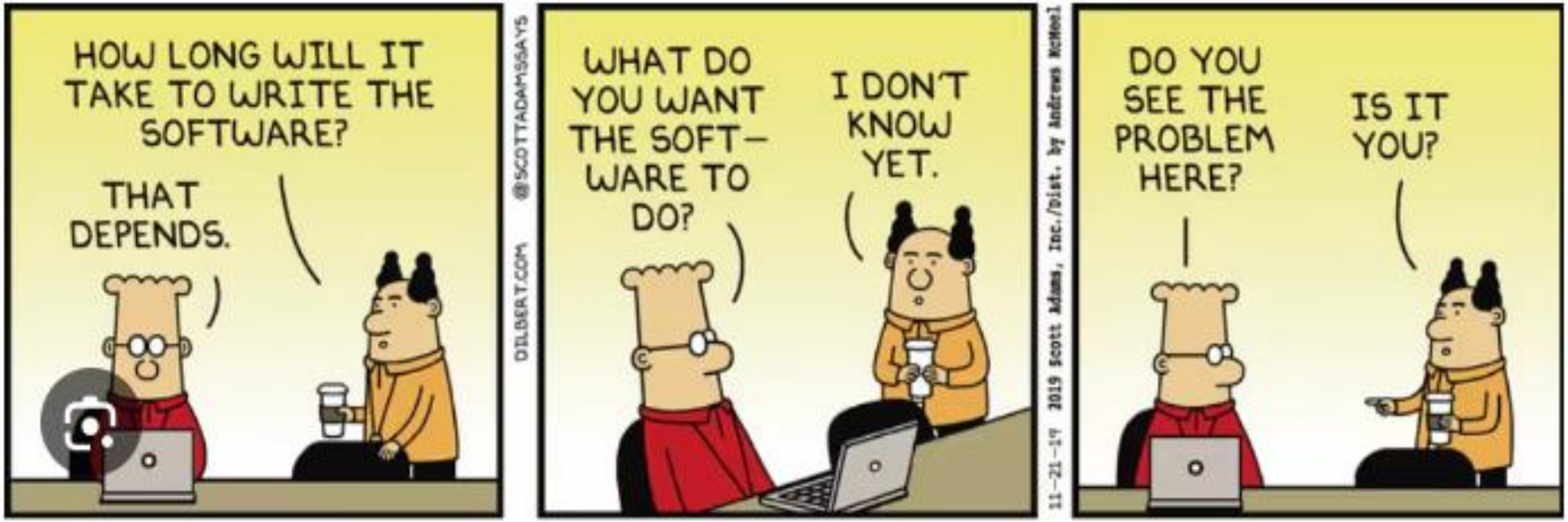
- Better and a shared understanding of needs

- Identify potential issues early on

- Gap analysis

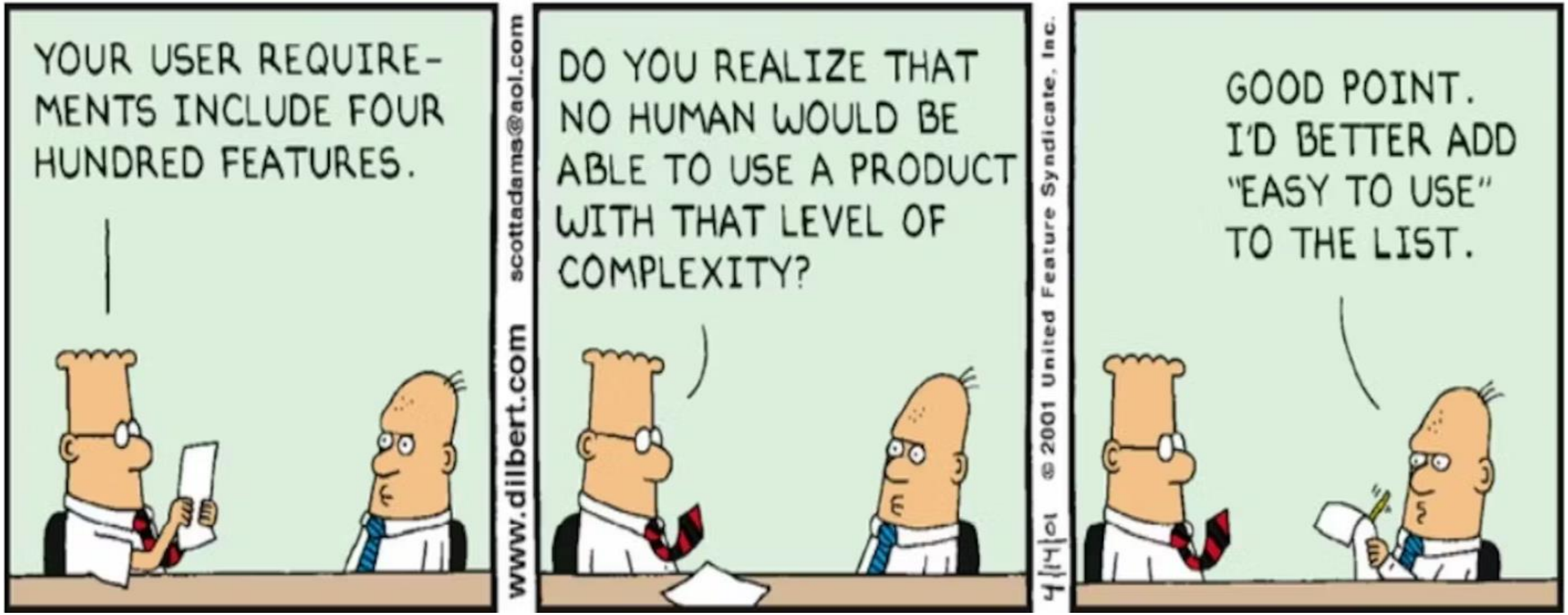
Requirements Engineering - Challenges

<u>Stakeholders do not know what they want!!</u>


HOW LONG WILL IT TAKE TO WRITE THE SOFTWARE?
THAT DEPENDS.
DILBERT.COM @SCOTTADAMSSAYS
WHAT DO YOU WANT THE SOFT-WARE TO DO?
I DON'T KNOW YET.
11-21-19 2019 Scott Adams, Inc./Dist. by Andrews McMeel
DO YOU SEE THE PROBLEM HERE?
IS IT YOU?

Requirements Engineering - Challenges

<u>Requirements are not precise, Opinions differ</u>


YOUR USER REQUIREMENTS INCLUDE FOUR HUNDRED FEATURES.
DO YOU REALIZE THAT NO HUMAN WOULD BE ABLE TO USE A PRODUCT WITH THAT LEVEL OF COMPLEXITY?
GOOD POINT. I'D BETTER ADD "EASY TO USE" TO THE LIST.
scottadams@aol.com
www.dilbert.com
© 2001 United Feature Syndicate, Inc.
4/14/01

Requirements Engineering - Challenges

Resolving conflicting views is time consuming

TODAY IS THE THREE-YEAR ANNIVERSARY OF OUR FIRST MEETING TO DISCUSS PROJECT REQUIREMENTS.
scottadams@aol.com
www.dilbert.com
AND WE'RE STILL DISCUSSING REQUIREMENTS. DOES ANYONE ELSE SEE A PROBLEM HERE?
4/4/02 © 2002 United Feature Syndicate, Inc.
WHEN YOU'RE DONE, CAN WE TALK ABOUT REQUIREMENTS?

Takeaways

- Poor requirements is one of the reasons why a project can fail.
- Documenting software requirements ensures that final product delivers the intended value
- Discussions facilitate creating clear requirements documents
- Verbose Requirements document aids in Improving Software Quality

Recap

- List stages of requirements Analysis?

- What are different ways of requirements elicitation?

- Who are stakeholders?

- What are tools for Requirements Modelling?

Thank You!!