

CSE 4513

Lec - 1

Introduction to SW Engineering



# IMPORTANT FOR YOU

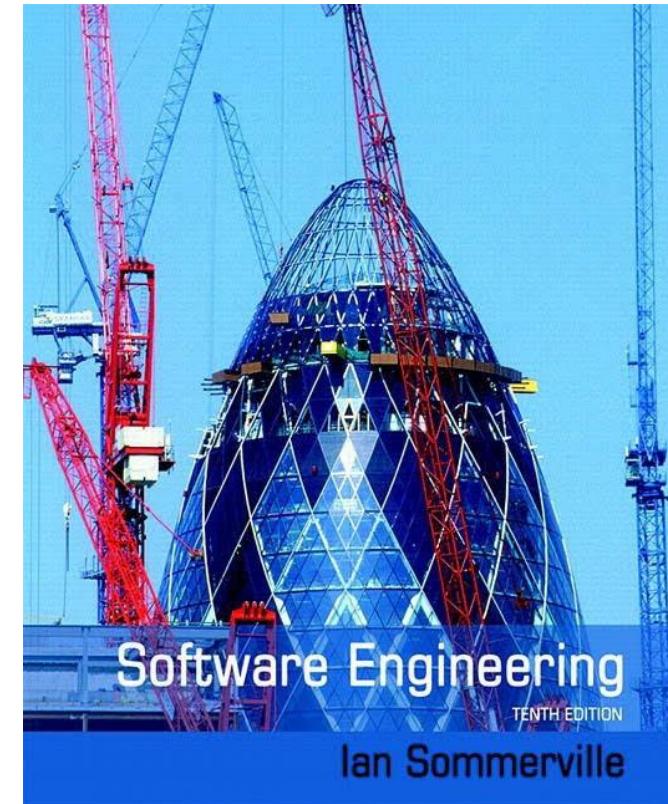
Course Code: CSE 4513

Course Name: Software Engineering and Object Oriented Design

Text Book:

1. Software Engineering a practitioner's approach, Author: Roger S. Pressman Fourth edition, McGraw-Hill Book Company
2. Object Oriented Modeling and Design, Author: Raumbugh
3. Unified Modeling system, Author: Raumbugh
4. Applying UML and Patterns, Author: Craig larman
5. Object oriented analysis and design Using UML Author: Bennet Farmer
6. Design Pattern, Author: Gamma, Helm & Jhonson

I will share what I did in  
Industry during last 10 years



# SHOHEL AHMED



## Professional Experience:



Islamic University of Technology (IUT)  
**Assistant Professor (Feb'2020 ~ )**



Military Institute of Science & Technology  
**Assistant Professor (Sep'19 - Present)**



Kona Software Lab Ltd.  
**Sr. Project Manager (Jul'16 ~ Sep'19)**



Samsung R&D Institute Bangladesh (SRBD)  
**Chief Engineer & Group Leader (Jan'15~ Jul'16)**  
*Digital Imaging*  
**Technical Lead (Dec'10~ Dec'14)**  
*Platform Lab*



Sylhet International University  
**Assistant Professor (Jul'09 – Nov'10)**  
**Lecturer (Apr'04 – Jul'04)**



Inha University, Korea  
**Research Assistant (Aug'07-Jun'09)**  
*Intelligent Mobile Lab*

## Educational Experience:



Master of Engineering in Information Engineering  
**Inha University, Incheon, South Korea**



BSc in Computer Science & Information Technology  
**Islamic University of Technology, Gazipur, Bangladesh**



Higher Secondary Certificate (HSC)  
Secondary School Certificate (SSC)  
**Rangpur Cadet College, Rangpur, Bangladesh**

## Patents:

- ✓ **Shohel ahmed, Shake Md. Riazul Islam, Kwak Kyung Sup - "METHOD AND SYSTEM FOR FINGERPRINT-BASED PAYMENT"** Application No.(Date) 1020170141375 (2017.10.27), Publication : KR101951125B1
- ✓ BARI, A.S.M. Hossain, **Shohel AHMED,AHSAN, Shegufta Bakht, Sadre Ala PARVEZ,AZIZ, Syeda Persia, Md. Tawhidul Islam CHOWDHURY-**  
**"ELECTRONIC DEVICE AND METHOD FOR PROVIDING INFORMATION THEREOF"**  
Publication number: 20150220247



# LOOKING FOR SW ENGINEER

## Jr. Software Engineer

### Job Responsibilities

- Understand properly SDLC and coordinate with other members of the team.
- Analyse the requirements and understand the deliverables
- Code according to software design specifications following standard coding styles and practices.
- Seek out new technologies and ideas to add value to project
- Participate in daily scrum meetings
- Participate in sprint planning
- Work closely with the Architecture Team to ensure architectural integrity and product quality.
- Work closely with the QA team, Product Management team, and the Research and Development manager to ensure quality and punctual software development.

### Additional Requirements

- Age 20 to 30 years
- Must have ACM problem solving skill.
- Those who have solved 1000+ ACM problems will get preference.
- Good score on HackerRank, Codeforces, Codechef, Light OJ and UVA Online Judge or any other problem solving sites will be considered as a plus.
- Must have strong knowledge of any major programming language, like C/C++, Objective-C or Java.
- Clear concept of Object Oriented Programming.
- Experience with MVC frameworks.
- Knowledge of algorithms and excellent problem-solving capability.
- Must be efficient at competitive programming.



# GOAL OF THIS COURSE

---

01

Help to develop **skills** that will enable to construct industry standard software of high quality.

02

To foster an understanding of why these **skills** are important.



# WHAT IS SOFTWARE(SW)

---

is a **set of instructions** (computer program) and its **associated documentations** that tells a computer **what to do** or **how to perform a task**.



# ATTRIBUTES OF GOOD SOFTWARE

| Product characteristic     | Description  |
|----------------------------|--|
| Maintainability            | Software should be written in such a way so that it can evolve to <b>meet the changing needs of customers</b> . This is a critical attribute because software change is an inevitable requirement of a changing business environment.  |
| Dependability and security | Software dependability includes a range of characteristics including reliability, security and safety. Dependable software <b>should not cause physical or economic damage in the event of system failure</b> . <b>Malicious users should not be able to access or damage the system</b> . |
| Efficiency                 | Software should not make <b>wasteful use of system resources such as memory and processor cycles</b> . Efficiency therefore includes responsiveness, processing time, memory utilisation, etc.   |
| Acceptability              | Software must be <b>acceptable to the type of users for which it is designed</b> . This means that it must be <b>understandable, usable and compatible</b> with other systems that they use.   |



# INDUSTRY STANDARD SW

An industry standard SW must be:

## Operational

- Budget
- Efficiency
- Usability
- Dependability
- Correctness
- Functionality
- Security

## Transitional

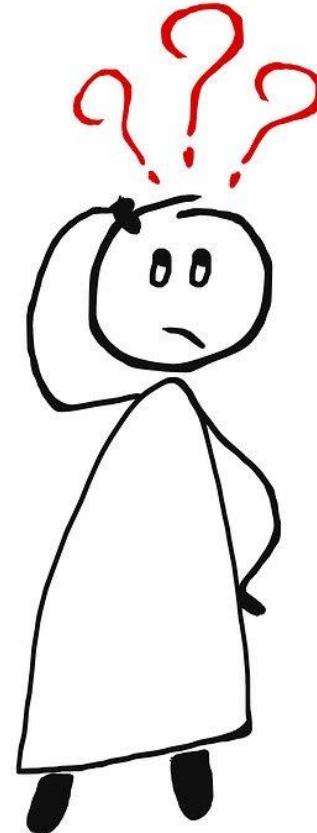
- Interoperability
- Reusability
- Portability
- Adaptability

## Easy to Maintain

- Flexibility
- Maintainability
- Modularity
- Scalability

# WHAT IS SW ENGINEERING

IEEE, in its standard 610.12-1998,  
Software engineering is defined as a  
systematic, disciplined, which  
**software, delivered on time** and  
**operation, and maintenance**.



as the application of a  
production of **quality**  
**development,**  
satisfies its **requirement.**



# SW DEVELOPER(PROGRAMMER) VS SW ENGINEER

According to Glassdoor.com, software developers (programmer) can expect to earn around **\$80,000** per year. As opposed to that, software engineers can expect a salary of **\$103,000** per year.



# KEY DIFFERENCE

---

- ✓ Software Engineer applies the principles of software engineering for designing, development, maintenance, testing, and evaluation of computer software whereas Software Developer builds software which runs across various types of computer.
- ✓ Software Engineer works with other components of the hardware system whereas Software Developers write a complete program.
- ✓ Software Engineer creates the tools to develop software while Software Developers use readymade tools to build apps.
- ✓ Software Engineer tends to solve issues on a much larger scale whereas Software Developers tend to do everything that engineers do but on a limited scale.



What is Industry demand?

**SW Developer or SW Engineer**



# CHALLENGES OF SOFTWARE ENGINEERING

---

- ✓ In safety-critical areas such as space, aviation, nuclear power plants, etc. the cost of software failure can be massive because lives are at risk.
  
- ✓ Increased market demands for fast turnaround time.
  
- ✓ Dealing with the increased complexity of software need for new applications.



# SOFTWARE ENGINEERING ETHICS

ACM/IEEE-CS Joint Task Force on Software Engineering Ethics and Professional Practices.

## The Ethical Principles are:

1. **PUBLIC** - Software engineers shall act **consistently** with the **public interest**.
2. **CLIENT AND EMPLOYER** - Software engineers shall act in a manner that is in the **best interests of their client and employer** consistent with the public interest.
3. **PRODUCT** - Software engineers shall ensure that their products and related modifications meet the **highest professional standards** possible.
4. **JUDGMENT** - Software engineers shall maintain **integrity and independence** in their professional judgment.
5. **MANAGEMENT** - Software engineering managers and leaders shall subscribe to and **promote an ethical approach** to the management of software development and maintenance.
6. **PROFESSION** - Software engineers shall advance the **integrity and reputation** of the profession consistent with the public interest.
7. **COLLEAGUES** - Software engineers shall be **fair to and supportive** of their colleagues.
8. **SELF** - Software engineers shall participate in **lifelong learning** regarding the practice of their profession and shall promote an ethical approach to the practice of the profession.  
<https://ethics.acm.org/code-of-ethics/software-engineering-code/>



# **SW Development Life Cycle (SDLC)**



# SW DEVELOPMENT LIFE CYCLE (SDLC)

---

- ✓ is a systematic process for building software that ensures the **quality** and **correctness** of the software built.
- ✓ SDLC aims to produce **high-quality** software that meets **customer expectations**.



# WHY SDLC

---

- ✓ It offers a basis for project planning, scheduling, and estimating
- ✓ Provides a framework for a standard set of activities and deliverables
- ✓ It is a mechanism for project tracking and control
- ✓ Increases visibility of project planning to all involved stakeholders of the development process
- ✓ Increased and enhance development speed
- ✓ Improved client relations
- ✓ Helps you to decrease project risk and project management plan overhead



# SDLC PHASES



Phase 1: Requirement gathering and analysis

Phase 2: Feasibility

Phase 3: Design

Phase 4: Coding

Phase 5: Testing

Phase 6: Installation/Deployment

Phase 7: Maintenance



# REQUIREMENT GATHERING AND ANALYSIS

- ✓ is one of the most critical phases of SDLC.
- ✓ the basis of the whole software development process.
- ✓ All the business requirements are gathered from the client in this phase.
- ✓ Requirements are analyzed to prepare SW specification, A well defined document known as Software Requirement Specification (SRS).

is divided into two phases where **requirement gathering** is related to **identify client's projects technical requisites** and the **Analysis** is to evaluate whether **the accumulated information and facts are valuable enough** to proceed with the project plan.



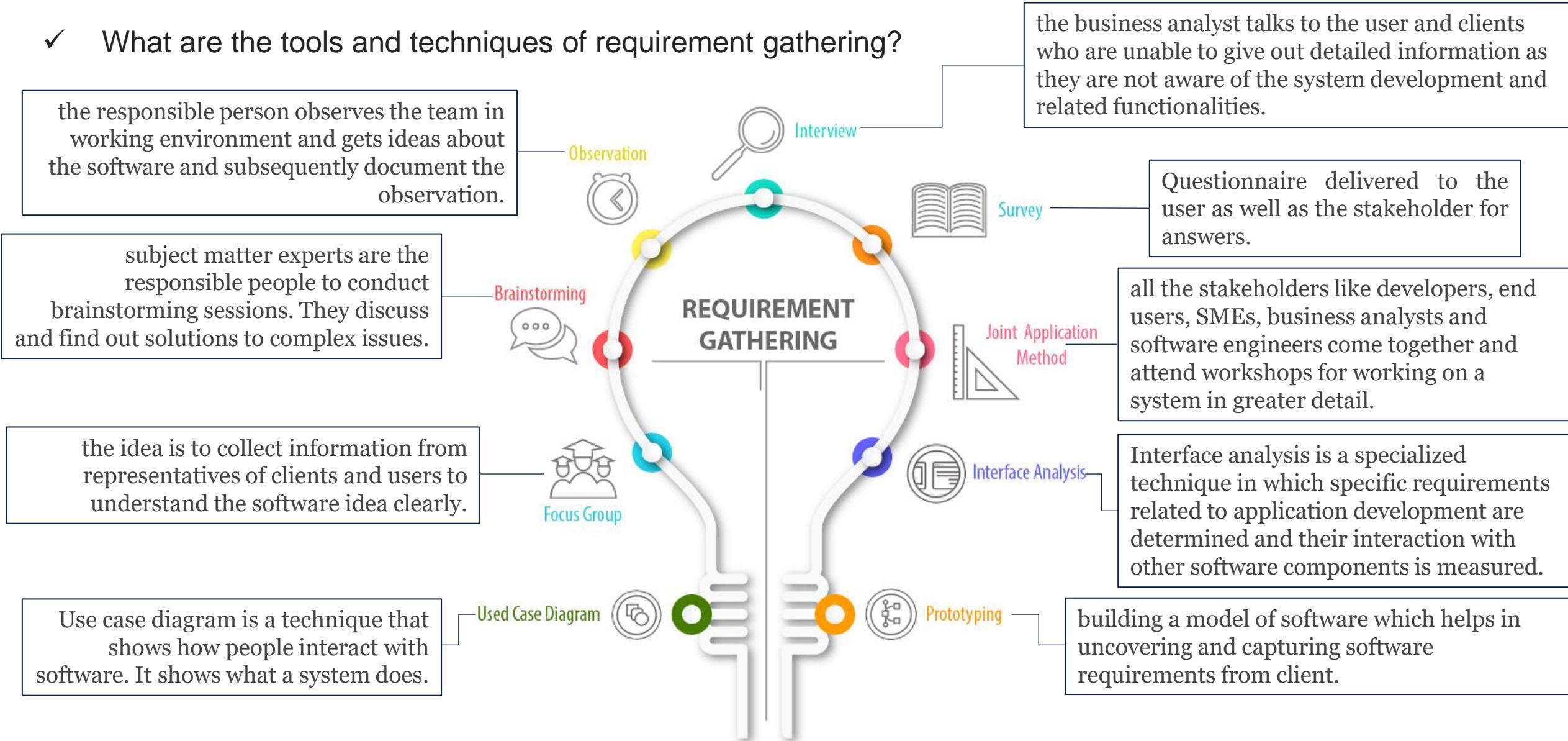
# REQUIREMENT GATHERING AND ANALYSIS

---

- ✓ Why is the requirement gathering important?
  - is all about eliciting project requisites from client end so that they can get a full-fledged product embedded with all the specifications.
  - it helps all the project associates communicate effortlessly with each other over defined clients needs.
  
- ✓ Who is responsible for requirements gathering?
  - **Business Analysts** are the professionals who efficiently carry out software requirement gathering by **breaking down the critical technical specifications into effective documentation and user stories.**

# REQUIREMENT GATHERING AND ANALYSIS

- ✓ What are the tools and techniques of requirement gathering?





# REQUIREMENT GATHERING AND ANALYSIS

## Challenges in Requirements Gathering

*Lack of Clarity  
in Defining  
Criteria for  
Success*

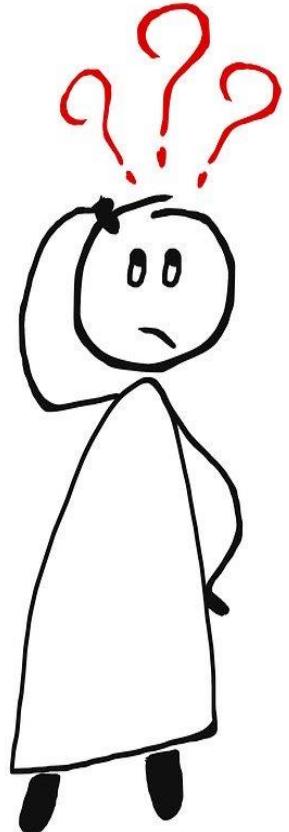
*Clients Change  
their Minds  
Often*

*Lack of or  
Over  
Communication  
by Clients*

*Clients get  
stuck over  
certain  
Techniques/Sol  
utions*

*Stakeholders  
can have  
Conflicting  
Priorities*

# REQUIREMENT GATHERING AND ANALYSIS



You are business analyst of XYZ Ltd. ABC bank wants to develop a mobile application of their banking products for their end users. Your company appointed you to collect the requirements for the mobile application from ABC Bank. **What technique will you use to collect the requirement and what are the answers you need to get from your stakeholders??**

# FEASIBILITY

## Five types of feasibility check we need to perform

### Operational

Can we create operations which is expected by the client?

03



### Technical

Need to check whether the current computer system can support the software

04



### Schedule

Decide that the project can be completed within the given schedule or not.

05



### Economic

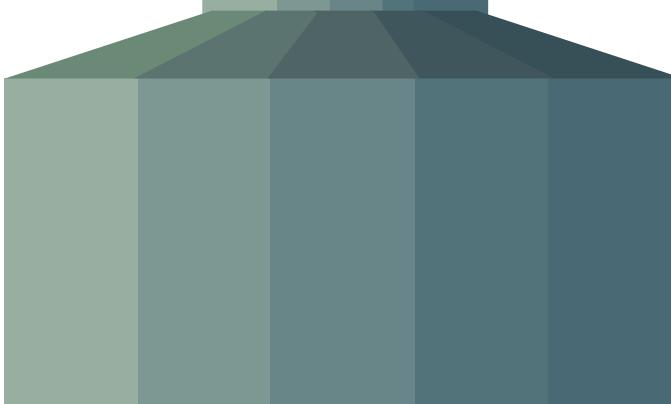
01

Can we complete the project within the budget or not?

### Legal

02

Can we handle this project as cyber law and other regulatory framework/compliances.





# DESIGN

---

- ✓ Design documents are prepared as per the SRS.
- ✓ Helps to define overall system architecture.

There are two kinds of design documents developed in this phase:

## High-Level Design (HLD)

- ✓ Brief description and name of each module
- ✓ An outline about the functionality of every module
- ✓ Interface relationship and dependencies between modules
- ✓ Database tables identified along with their key elements
- ✓ Complete architecture diagrams along with technology details

## Low-Level Design(LLD) / Detail Level Design (DLD)

- ✓ Functional logic of the modules
- ✓ Database tables, which include type and size
- ✓ Complete detail of the interface
- ✓ Addresses all types of dependency issues
- ✓ Listing of error messages
- ✓ Complete input and outputs for every module

# DESIGN

## Why software design is important?



### Modularity

When you have a modular software, it's easy to move or even remove things if needed and even share the design work between multiple developers per software component.



### Performance

design should tell how your system works internally and how it uses resources such as threads, database connections and other things that might make a hit to the performance.



### Usability

design works as a great starting point for new-comers. Instead of making them look through all the source files you can just point them to the software design where they find all information they'll need to get started.



### Maintainability

With good software design it's easier to maintain the software. You can spot straight away from the design **how much a bug fix or introduction of a new feature will change the existing code base.**



### Portability

By including dependencies to other software modules, such as 3rd party libraries, in to your design makes the software much easier to port into another environment.



### Trackability

Good design **tracks the requirements** and proves on the design level that all that is required from a piece of software is actually there..



# CODING

---

- ✓ developers start build the entire system by writing code using the chosen programming language.
- ✓ tasks are divided into units or modules and assigned to the various developers.
- ✓ is the longest phase of the Software Development Life Cycle process
- ✓ developer needs to follow certain predefined coding guidelines
- ✓ also need to use programming tools like compiler, interpreters, debugger to generate and implement the code.



# CODING

---

Things which we need to take care in coding phase:

1. Version control
2. Before actual coding, you must spend sometime to select dev tool which will be suitable for debugging, coding, modification.
3. before actual writing code, some standard should be defined, as multiple developers going to use the same file for coding
4. during development, developer should write appropriate comments so that others will come to know the logic behind the code
5. there should be a regular review meeting need to conduct in this stage to identify the prospective defects in an early stage.



# TESTING

---

- ✓ Focus on investigation and discovery
- ✓ defines how the SW will be tested and who will perform
- ✓ find whether the developed code work according to the customer requirement.
- ✓ QA and testing team may find some bugs/defects which they communicate to developers.
- ✓ development team fixes the bug and send back to QA for a re-test.
- ✓ process continues until the software is bug-free, stable, and working according to the business needs of that system.



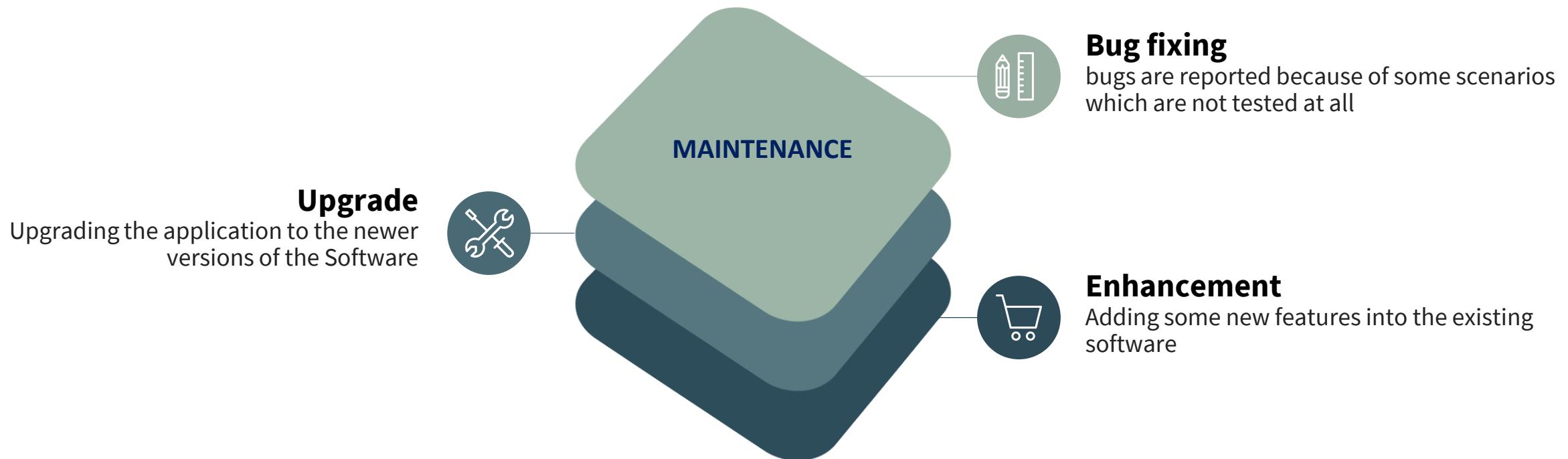
# INSTALLATION/DEPLOYMENT

---

- ✓ Deployment is the process of promoting a fully-tested and approved application
- ✓ **QA manager** finalized the final bug-free release items.
- ✓ Deployment guide is provided to the operation team for hassle free deployment
- ✓ Deployment could require different **promotion processes**
  - ✓ For web applications, it would require promoting the application to a server, testing, redirecting DNS
  - ✓ For standalone applications, it would require producing media or downloadable packages
  - ✓ **For software as a service, there are licensing issues that need to be determined**
- ✓ Ensure data migration to the live server has occurred completely

# MAINTENANCE

Major Activities after SW deployment/while SW in use by the end users



The main focus of this phase is to ensure that needs continue to be met and that the system continues to perform as per the specification mentioned in the first phase.



Why do we need a SW upgrade?



## **SW Development Life Cycle (SDLC) Models**