What is SDLC

- Framework / Process / Model / Guideline / Approach.
- describes activities performed at each stage of software development project
- Defined steps to software project

Benefits:

1. Clear Goals & Responsibility
2. Project Tracking & visibility.
3. Improved quality product.
4. Decrease proj. mgmt overhead.

1. Analysis & Planning

- Gather Business requirements from clients / stakeholders
- Evaluate feasibility of product / project.
  - cost of production
  - revenue potential
  - Technical / operational / economic / legal / schedule feasibility.
- Resource Allocation → tools, people, scheduling
- Scope of work (SoW) : High level document
  - details what is being built
  - what will be seen at end of proj.

↓

- Project specs
- Deliverables
- Timelines. →
  - Analysis & Planning (1 - 3 wks)
  - Requirements Gathering (1 - 2 wks)
  - Design & Prototyping (2 - 4 wks)
  - Software Development (4 - 6 wks)
  - QA Testing (1 - 2 wks)
  - UAT Testing (1 - 2 wks)
  - Deployment (1 wk)
  - Maintenance & updates (repeat) (Hypercare period 1 wk → Hotfix if required)
- Agreement

* Timeframe may be days or months depending on scale & nature of project.

2. Requirements

- Gathering requirements from business users.
  - schedule meeting / workshops
  - Share template to business user to fill & return.

Stakeholders for meeting: (Brainstorming)

- Business team (clients)
- Business Analyst
- Project manager
- Technical Architect
- Lead Developer

What to discuss in meeting :

- How existing system work / any existing system in place
- Who is going to use app (actors)
- What input / output required
- 3rd party APIs / tools
- Security & privacy requirements
- Possible limitations.

* Should not immediately begin coding after this

Next steps :

1. Business Analyst prepare Software Requirements Specification Document (SRS / user stories if Agile)

(1) └ Functional & Non functional requirements

- What the software should do.

eg: • Login form for user
• Registration form

- Quality constraint
- How software should behave

eg: • Performance / Response time
• Accessibility • Availability
• Scalability • Maintainability

- Every page should load in 3s
- App should be accessible over browser & mobile
- Available 99% of time

* important for this SRS to be detailed

- Business users / clients need to review SRS & agree.

(2) Technical Requirements

- Tech Stack
- database

} Tech architect will provide

* see sample document in selflearning SDLC dir.

└ can add sections as required.

3. Design & Prototyping

- SRS document (input) reference → wireframes + prototypes (output) – using figma / canva etc.

(Architect)
High-Level Design

- shows entire system architecture
- Database Design
- Data Flow diagram
- Services / Platforms / modules used (Brief description)

(lead developer)
Low-level Design

- Each module design & specificatn

eg: • UI design patterns to follow
• which classes to be created

- Detailed description of every module
- UI , Backend , DB module

- customer / Business team will see how final product looks like (Prototype)
  - ↳ feedback gathering / doubts clarification.

Env : Dev : for devs coding
QA : for testers.
STAGE : for Biz user test
PROD : final env.

Wire frame

- idea of how UI looks like.
- Dev reference to create UI
- Done by designer (if any)

Adobe XD
Invision studio
Figma
Draw . io

4. Software Development

- Refer to SRS & model design for coding

Developer Manual : Good practices, naming conventions

- OOP + Design Patterns usage.
- change log summary document may be required per development
- output of this phase is working requirements feature
  - ↳ Dev env for initial test → QA / SIT for testers.

5. Testing

- most Impt phase in SDLC
- Quality Assurance testing by test team
  1. Functional testing : All features working as expected
  2. Integration testing : 3rd party tools / API working seamlessly
  3. Performance testing :
  4. Load testing : Simulate multiple users using
  5. Penetration testing : Simulate cyber attack.

  } Non functional testing
- After QA / SIT → send to UAT for clients to test
- After UAT → deployment to PROD.

6. Deployment

- Typically automated with Azure Dev Ops , Jenkins etc.
- By Dev Ops team (if any exist) , Detailed steps on how to deploy + backout / roll back plan
  1. RFC raised (Request for Change)
  2. RFC to be approved by change manager
  3. App deployed for users + hypercare period (~15 days) , hotfix - small deployment for fixing

7. Maintenance & Updates

- Bugs found by users. (Bug Fix)
- New features request (Feature)
- Changes to existing functionality (Improvement)

} Product Manager handle

* Product management team meet with Business team to prioritize

└ Remedy or ServiceNow → 1. Biz user log ticket with priority & Service Level Agreement
(Critical = 3hrs
High = 1 day
Medium = 5 days)

- App updates
- Server updates
- OS updates
- App framework updates

} entire SDLC rerun