Project Delivery Lifecycle

MDA402 Project Management

Josef Spurný
jspurny@mail.muni.cz

Faculty of Informatics, Masaryk University

December 12, 2024

Lecture Overview

1. Software Development Lifecycle
    - Definition
    - Stages
2. Development Lifecycle Models
    - Waterfall
    - Spiral
    - Agile
    - Big bang

SDLC

Definition

Definition 3.1

Project lifecycle is the process from beginning of the project to an end.

Software development lifecycle (SDLC):

- is the structured process used to **design**, **develop** and **test** high-quality software.
- is methodology that provides entire procedure of software development **step-by-step** [1]

SDLC

Definition

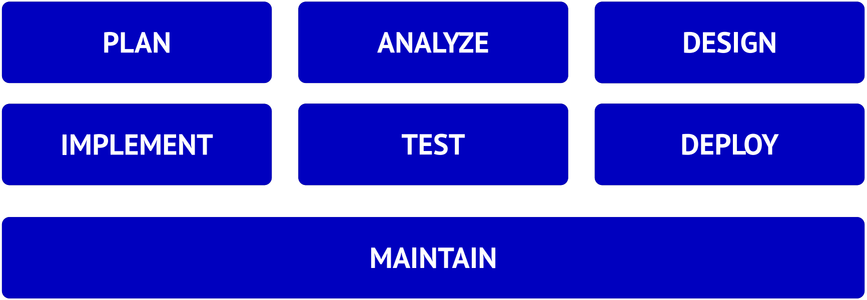
Goals of SDLC are:

- to provide **cost-effective** and **time-efficient** process for software development
- **minimize** project risks
- deliver **maintainable** software that meets **users' requirements**
- increase **visibility** of the development process for all stakeholders involved [2]

SDLC

Stages

... is methodology that provides entire procedure of software development **STEP-BY-STEP** → SDLC model operates in **seven** stages (may vary based on each company's view):



PLAN

ANALYZE

DESIGN

IMPLEMENT

TEST

DEPLOY

MAINTAIN

SDLC

Plan

- this is initial stage of SDLC → it's **pre-project** stage not always included in SDLC
- can be considered as the **most crucial** in terms of future success of the software development
- this stage contains these processes:
  - **define project scope**
  - **set goals and objectives**
  - **high-level requirements gathering**
  - **resource planning and allocation**
  - **establishing preliminary project schedule**
- usually outcome is called **software requirement specification** → document consisting of all relevant information agreed and gathered in this stage

SDLC

Analyze

- this phase contains **analysis** of gathered requirements in the previous phase
- it describes **WHAT** will be the delivered
- goals of this stage are:
  - **understand** needs and expectations of key stakeholders
  - **define** priorities
- analysis should be **high-level** → avoid going into too much details (done in next stage)

SDLC

Design

- based on the outcome of the analysis stage, **solution documentation** is created
- it describes **HOW** product / service will be delivered
- this document is a guide for creating a new software (or software update) from analyzed requirements
- part of this stage is also the process of **reviewing** and **signing off** the solution document by all involved stakeholders

SDLC

Implement

- this phase contains **implementation / build / development** of software
- the build is done based on the solution document created in the previous stage
- **individual functional modules** are built in this stage → **translation** of solution documentation into functional modules
- in this stage software / service begins to:
  - take **shape**
  - become **tangible** product

SDLC

Test

- this phase contains **testing** of the software implemented / built / developed
- currently used are these two types of testing: **automatic** & **manual**
- it can also run in **parallel** to other phases → e.g. during Implementation phase the code is tested immediately after being written
- visualization of testing phase and its layout through whole SDLC → **V-MODEL**

SDLC

Test

Definition 3.2

V-model is a structured model, where each phase of this SDLC is integrated with testing phase. It's executed in sequential manner. It's known as verification and validation model.

- clearly illustrates that testing is **not dependent** on coding
- testing process starts at the beginning
- design of test cases can start with analyzed requirements and identified use cases → **use case = test case**

SDLC

Test

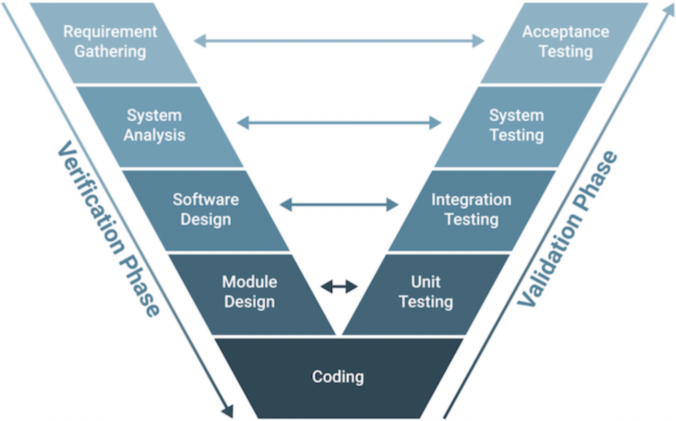


Figure: V-model visualization [3]

SDLC

Test

Types of testing explained:

- **unit** testing → focused testing on small code units (mostly done by developers during implementation)
- **system** testing (ST) → new functionality testing
- **system integration** testing (SIT) → integration testing of new functionality (with old system or third parties involved...)
- **user acceptance** testing (UAT) → new functionality testing by the client / end-user
- **regression** testing → core & new functionality testing before deployment
- **smoke** testing → core functionality testing after deployment

SDLC

Deploy

- this phase contains **final deployment** of the implemented software into **production**
- production is the version of software that client uses as a **product**
- based on the SDLC model there could be several deployments into production
- implementation and testing are done in so-called test environment

SDLC

Maintain

- this is final stage of SDLC → it's **post-project** stage not always included in SDLC
- contains **maintenance** of the deployed software
- maintenance can be understood as:
  - product **operation** by admin user → this contains some manual activities that needs to be regularly carried out
  - software **monitoring** for potential unresolved issues, unidentified bugs, performance
  - **managing change** and **fixing** of mentioned issues and bugs

Development Lifecycle Models

Definition 3.3

Development lifecycle model presents SDLC in specifically organized fashion to help with its implementation and to suite project needs.

Currently there exists more than 50 SDLC models, we will look at these models:

WATERFALL

SPIRAL

AGILE

BIG BANG

Development Lifecycle Models

Waterfall

Definition 3.4

Waterfall model is based on sequential execution of phases, where each subsequent phase depends on the outcome of previous phase.

Pros

- disciplined project management
- easy to understand, simple to use
- suitable for small projects with clear requirements

Cons

- no room for change
- product can be used only after the whole lifecycle is completed
- not suitable for software development

Development Lifecycle Models

Waterfall

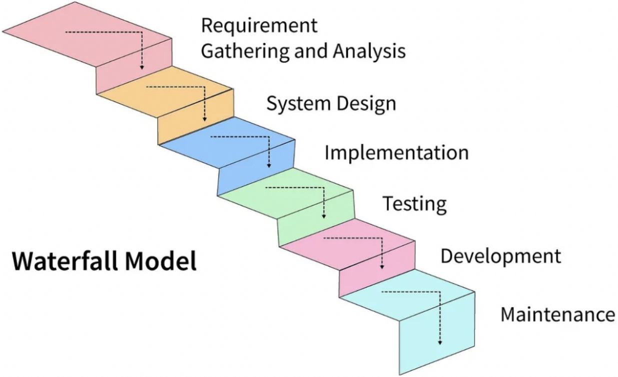


Figure: Waterfall visualization [4]

Development Lifecycle Models

Spiral

Definition 3.5

Spiral model adds iterative element (small repeated cycles) to the sequential flow of waterfall model.

Pros

- it's incremental → small pieces delivery
- can handle changes & supports risk handling
- suitable for SW development

Cons

- expensive for smaller projects
- indefinite continuation of the spiral
- intermediate stages required

Development Lifecycle Models

Spiral

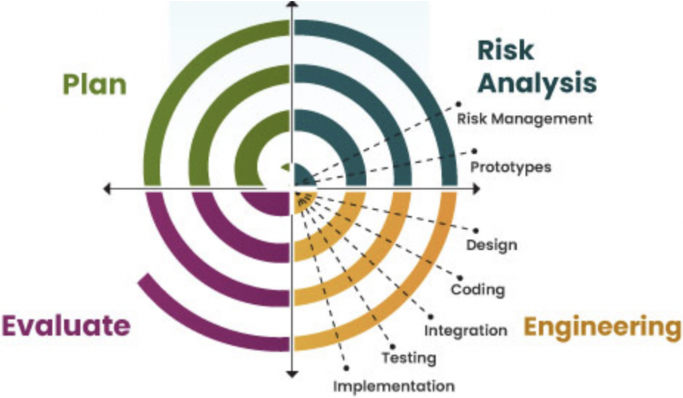


Figure: Spiral visualization [5]

Development Lifecycle Models

Agile

Definition 3.6

Agile model is a variation of spiral. Agile iterates rapidly through stages delivering software incrementally.

Pros

- change is welcomed & quickly responded to
- product is ready to use during lifecycle
- high client involvement
- most suitable for SW development

Cons

- vulnerability to excessive scope changes
- limited documentation can lead to uncertainty
- not suitable for small projects
- hard to measure upfront

Development Lifecycle Models

Agile

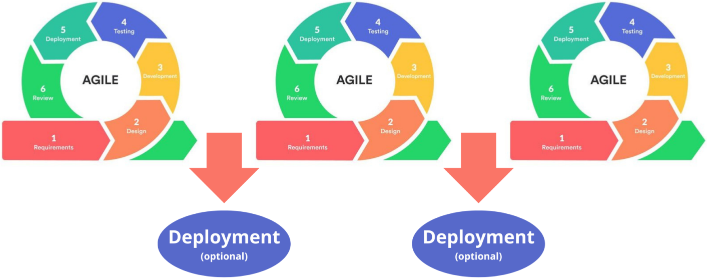


Figure: Agile visualization [6]

Development Lifecycle Models

Big bang

Definition 3.7

Big bang model is informal and unstructured SDLC model with no planning, documentation or defined phases.

Pros

- simple model to use
- little or no planning & few resources required
- flexible for developers

Cons

- high risk in uncertainty
- not suitable for complex projects
- can be too expensive

Development Lifecycle Models

Big bang

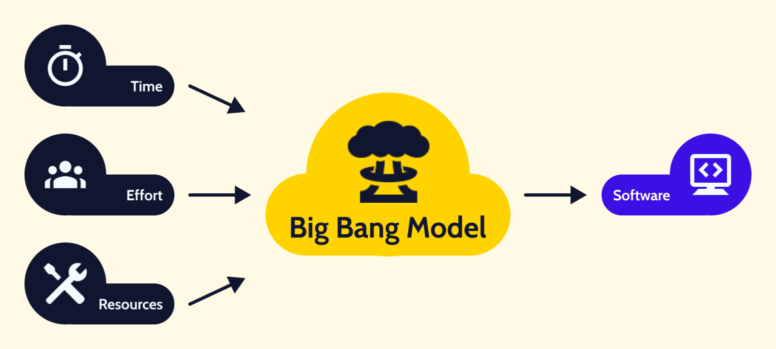


Figure: Big bang visualization [7]

Bibliography I

[1] "Software Development Life Cycle (SDLC)", 2024. [Online]. Available: https://www.geeksforgeeks.org/software-development-life-cycle-sdlc/. [Accessed: 24-Sep-2024].

[2] "What is SDLC (Software Development Lifecycle)?", 2024. [Online]. Available: https://aws.amazon.com/what-is/sdlc/. [Accessed: 24-Sep-2024].

[3] "What Is the V-Model in Software Development?", 2023. [Online]. Available: https://builtin.com/software-engineering-perspectives/v-model. [Accessed: 25-Sep-2024].

Bibliography II

[4] "The Waterfall Model: A Linear Approach to Software Development", 2023. [Online]. Available: https://medium.com/@ksandhya/the-waterfall-model-a-linear-approach-to-software-development-73bb48e154b4. [Accessed: 25-Sep-2024].

[5] "Spiral Model For Software Development- A Risky-Driven Model", 2023. [Online]. Available: https://www.bdtask.com/blog/spiral-model-for-software-development. [Accessed: 25-Sep-2024].

[6] "Agile Methodology: Meaning, advantages, disadvantages & more", 2024. [Online]. Available: https://targettrend.com/agile-methodology-meaning-advantages-disadvantages-more/. [Accessed: 25-Sep-2024].

Bibliography III

[7] "Big Bang Model", 2023. [Online]. Available: https://www.codecademy.com/resources/docs/general/software-development-life-cycle/big-bang-model. [Accessed: 25-Sep-2024].

Thank You for Your Attention!