Reading_week3

试用水印

Top 6 software development models and how they influence the SDLC

- What is Software Development Life Cycle?
  - The Software Development Life Cycle (SDLC) **comprises all stages of software production — from analysis to deployment and support**, and it will differ depending on the software engineering model you follow.
  - Software Development Phases
    - 1. Planning (requirements analysis)
      - The software delivery and business teams **collect information to outline the product’s business objectives.**
      - The requirements-gathering process helps to **align business objectives with the product** being developed.
    - 2. Prototype design
      - the software delivery team defines how the project will be developed and reflects this information in the **Design Document Specification (DDS)**
      - this documentation is made available to all parties involved in the development process (developers, testers, managers, and the client).
    - 3. Development (implementation)
      - This is the core phase of all software process models, during which engineers **develop the solution**(code-writing, fine-tuning) according to the requirements.
    - 4. Quality assurance (QA) and testing
      - This phase consists of **code reviews and module assessment**
      - refer to unit and integration testing
    - 5. Delivery
      - The project manager, the client, and the stakeholders have to **approve the project.** After they do, it’s time for the **product to hit the market.**
    - 6. Maintenance
      - most applications require **regular maintenance** to remain relevant
      - This phase covers compatibility upgrades, performance optimization, and monitoring, as well as bug fixes.
- Why following the SDLC process is important
  - Sticking to the SDLC process of your software development model will help you deliver the project on schedule and within budget.
  - It helps the project stay on track.
  - It maximizes productivity.
  - It reduces development time.
  - It provides better management.
- Types of SDLC models
  - The Iterative model

    - Iterative software process models **begin with a smaller set of requirements**. Developers use these **requirements to** analyze and gradually implement **features**.
      Afterwards, these features are evaluated and tested **to identify new requirements.**
    - suitable for
      - Large-scale projects with multiple modules (such as web services or microservices)
      - Projects with clearly defined objectives and tasks
  - The Agile model

    - Agile focuses on continuous release cycles and cross-functional development
    - However, unlike Iterative, Agile doesn’t rely on the baseline plan.
      The plan, along with the requirements, is continuously modified throughout the life cycle.
    - Suitable for
      - Large-scale and smaller projects
      - Outsourcing and managed IT services
      - Adding new features to a working prototype
  - The DevOps Model

    - DevOps is built around the **principles of automation and collaboration**.
      This model’s primary goal is to enhance cooperation **between the operations team and developers** (hence, its name) thanks to continuous feedback.
    - The model shares many similarities with Agile, but there’s one fundamental difference: **DevOps** focuses on **interactions between the technical departments and managers**. **Agile**, on the other hand, **emphasizes direct client (or product owner) involvement**.
    - Suitable for
      - Complex projects that require a lot of QA and testing
      - Large teams with multiple departments
  - The Waterfall Model

    - Waterfall is one of the oldest types of software development life cycle methodologies.
      It focuses on **a linear approach** with SDLC phases following one another sequentially.
    - Suitable for
      - Smaller and mid-sized projects with clearly defined requirements
  - Verification and validation (V-shaped)

    - Verification and Validation is **a sequential methodology**, just like Waterfall.
      But there’s a **difference** — in this methodology, **every step goes in parallel with testing**.
    - Suitable for
      - For mid-sized and large projects with explicit objectives and requirements
  - The Spiral model

    - Spiral combines the **Iterative and Waterfall methodologies** with an emphasis on risk assessment.
    - Suitable for
      - Larger projects with complex requirements
      - New products with multiple testing stages

IT Project Management: Managing IT Projects

- What Is IT Project Management?
  - IT project management (ITPM) is the planning, scheduling, execution, monitoring and reporting of IT projects.
- What Are the Six Phases of an IT Project?
  - Initiation
    - During the first phase of an IT project, one must ask: “**why is this project needed?**” —in other words, the **objective of the project** must be identified.
  - Definition
    - This is where the **objectives of the project are finalized** and the **requirements** for a successful project are **identified**.
  - Design
    - The design phase of an IT project is when the project team sets out to **find the best solution for achieving their goal.**
  - Development
    - The development phase is when the development team is **assigned tasks** and **project management tools are selected**.
  - Implementation
    - The implementation phase is where the **final deliverable of the IT project is developed**
  - Follow Up
    - The follow up phase is all the work that comes after the project is delivered, and includes **setting up support teams**, training the end-users, creating a postmortem and ultimately ending the project.
- What Does an IT Project Manager Do?
  - An IT project manager is responsible for **overseeing** an organization’s IT department and **managing** teams to execute IT projects on time and **within budget**.

IT project management

- Difference Between Waterfall vs Spiral and Incremental Model
  - 看网页
- The IT project management lifecycle
  - ITPM follows the five phases of the project management lifecycle and is often managed using the Waterfall methodology
  - The most common ITPM method is the **Waterfall methodology**, which involves a predictive linear process.
  - Another lifecycle method is the **iterative method**, which uses a more incremental approach
  - IT project management may also use an **adaptive lifecycle**, such as those found in Agile methodologies.
- Challenges faced by IT project managers
  - Making multiple assumptions when integrating different hardware, networks, and software to the existing system
  - **Unclear expectations** from the business, end-users, and stakeholders
  - Rapidly changing technology, leading to **necessary mid-project upgrades** that can affect timelines
  - **Geographically diverse offices** and remote work associated