**Software Development Notes**

**Software Development Life Cycle (SDLC)**

The Software Development Life Cycle (SDLC) describes the steps software development teams follow to create high-quality software. It involves a structured series of phases involving developers, client-facing roles, and project managers.

**Two Main Types of SDLC Approaches:**

* **Waterfall Model (Structured Approach)**
* **Agile Model (Iterative Approach)**

**Waterfall Model**

A linear and sequential approach to software development where each phase must be completed before moving on to the next.

**Phases of the Waterfall Model:**

1. **Gathering Requirements:**
   * Conduct interviews with the client, stakeholders, and end users.
   * Write detailed requirement reports.
   * Establish key factors:
     + **Time:** Project timeline estimates.
     + **Cost:** Budget planning.
     + **Technical Requirements:** Define operating system (OS) compatibility and software functionalities.
2. **Design Software:**
   * **UI Design:** Create user interface designs.
   * **Visual Design:** Use flowcharts to map processes.
   * **Logical Design:** Develop algorithms to define software logic.
3. **Programming/Development:**
   * Use Git for version control.
   * Select the most suitable programming language.
   * Collaborate in a team-based environment.
4. **Testing:**
   * Check software functionality.
   * Detect and fix bugs.
5. **Deployment:**
   * Release the software, considering methods, timing, and phased rollouts.
6. **Maintenance:**
   * Ensure compatibility with OS updates and expanding platforms.
   * Add new features.
   * Perform bug fixes and accommodate changing requirements.

**Pros and Cons of the Waterfall Model:**

**Pros:**

* Well-defined project phases.
* Clear documentation.
* Easier to estimate timelines and costs.
* Reduced scope creep.

**Cons:**

* Requires significant upfront planning.
* Difficult to make changes once a phase is completed.
* Higher risk due to limited flexibility.

**Who Prefers the Waterfall Model?**

* **Government Organizations (e.g., ATO)**
* **Large Organizations (e.g., Banks)**

**Reason:**

* Provides a more defined structure.
* Ensures reliability and comprehensive documentation.

**Agile Model**

**“Sprints”**

* Dev teams work in short weekly development sprints. At the end of each sprints, they meet together, assess progress, show client and GO AGAIN!
* Daily check in

**Requirements**

* Fun
* Collectable

**Specifications**

* Specific targets

**Prototypes**

* Iterative versions

**Continuous testing**

* Constant testing to fit errors

**Full-Stack Developer Roles:**

A full-stack developer handles both front-end and back-end development along with database management.

**Key Components:**

1. **Front-End (User Interface):**
   * Technologies: HTML, CSS, JavaScript
2. **Back-End:**
   * Technologies: Python, PHP, JavaScript
   * Purpose: Enable applications to access and manage dynamic data.
3. **Database Layer:**
   * Technology: SQL Database

**Oli nunn’s phone number: *0417510539***

**Call this number ;)**