WEB DEVELOPMENT DAILY CHALLENGE

1. **Explain the following agile methodologies Scrum, Kanban, Extreme Programming.**

Answer

Kanban

Kanban is a scheduling system for lean and other JIT processes. In a Kanban process, there are physical (or virtual) “cards” called Kanban that move through the process from start to finish. The aim is to keep a constant flow of Kanban so that as inventory is required at the end of the process, just that much is created at the start. When used for software development, Kanban uses the stages in the software development lifecycle (SDLC) to represent the different stages in the manufacturing process. The aim is to control and manage the flow of features (represented by Kanban cards) so that the number of features entering the process matches those being complete. Kanban is an agile methodology that is not necessarily iterative. Processes like Scrum have short iterations that mimic a project lifecycle on a small scale, having a distinct beginning and end for each iteration. Kanban allows the software to be developed in one large development cycle.

Scrum

Scrum is an agile project management methodology or framework used primarily for software development projects with the goal of delivering new software capability every 2-4 weeks. It is one of the approaches that influenced the Agile Manifesto, which articulates a set of values and principles to guide decisions on how to develop higher-quality software faster. Scrum addresses complexity in work by making information transparent, so that people can inspect and adapt based on current conditions, rather than predicted conditions. This allows teams to address the common pitfalls of a waterfall development process: chaos resulting from constantly changing requirements; underestimation of time, resources and cost; compromises on software quality; and inaccurate progress reporting. Transparency of common terms and standards is required in Scrum development to ensure that what is being delivered is what was expected. A frequent inspection ensures progress and detects variances early on so that adjustments can be made quickly.

Extreme Programming

Extreme Programming (XP) is an agile software development framework that aims to produce higher quality software and higher quality of life for the development team, and also the responsiveness to changing customer requirements. As a type of agile software development, it advocates frequent "releases" in short development cycles, which is intended to improve productivity and introduce checkpoints at which new customer

requirements can be adopted. XP is the most specific of the agile frameworks regarding appropriate engineering practices for software development.
The five values of XP are communication, simplicity, feedback, courage, and respect

2. **Who are the members of an agile team and what are their roles?**
   Answer
   Team Lead, Scrum Master (Scrum), Team Coach, or Project Lead
   Acts as the coach responsible for facilitating and guiding the team, obtaining resources when required, and removing impediments that keep the team from doing their work. The Scrum Master role often encompasses the soft skills of project management more than planning and technical skills, which are often left to the team as a whole. It is important to note that this person is not necessarily the team's manager. Rather, this role should reflect knowledge and responsibilities over rank.

   Team Member
   Responsible for the project's creation and delivery. The team members will normally be comprised of developers, QA, and documentation. They are responsible for planning, design, development, testing, and project delivery.

   Product Owner (Scrum), On-Site Customer (XP), Active Stakeholder
   Represents the voice of the customer and is responsible for the prioritized backlog and maximizing the return on investment (ROI). Part of this role's responsibility includes documenting user stories or requirements for the project.

   Stakeholders
   Represents a broad category of people who can be users, managers of users, operations, support, Portfolio Managers, other Agile teams with dependencies, executive team, investors, and more.

3. **List 5 Project Management Tools.**
   Answer
   Slack
   Kabanize
   Trello
   Jira
   Bridge24

4. **List the phases of the Software Development Life Cycle.**
   Answer

1. **Planning**
   This is the first phase of the systems development process. It identifies whether or not there is a need for a new system to achieve a business"s strategic objectives. This is a preliminary plan (or a feasibility study) for a company"s business initiative to acquire the resources to build on an infrastructure to modify or improve a service. The company might be trying to meet or exceed expectations for its employees, customers, and stakeholders too. The purpose of this step is to find out the scope of the problem and determine solutions. Resources, costs, time, benefits and other items should be considered at this stage.

2. **Systems Analysis and Requirements**
   The second phase is where businesses will work on the source of their problem or the need for a change. In the event of a problem, possible solutions are submitted and analyzed to identify the best fit for the ultimate goal(s) of the project. This is where teams consider the functional requirements of the project or solution. It is also where system analysis takes place—or analyzing the needs of the end-users to ensure the new system can meet their expectations. Systems analysis is vital in determining what a business"s needs are, as well as how they can be met, who will be responsible for individual pieces of the project, and what sort of timeline should be expected.

   There are several tools businesses can use that are specific to the second phase.
   They include:
   CASE (Computer Aided Systems/Software Engineering)
   Requirements gathering
   Structured analysis

3. **Systems Design**
   The third phase describes, in detail, the necessary specifications, features, and operations that will satisfy the functional requirements of the proposed system which will be in place. This is the step for end-users to discuss and determine their specific business information needs for the proposed system. It"s during this phase that they will consider the essential components (hardware and/or software) structure (networking capabilities), processing and procedures for the system to accomplish its objectives.

4. **Development**

   The fourth phase is when the real work begins—in particular, when a programmer, network engineer and/or database developer are brought on to do the major work on the project. This work includes using a flow chart to ensure that the process of the system is properly organized. The development phase marks the end of the initial section of the process. Additionally, this phase signifies the start of production. The development stage is also characterized by installation and change. Focusing on training can be a huge benefit during this phase.

5. **Integration and Testing**

   The fifth phase involves systems integration and system testing (of programs and procedures)—normally carried out by a Quality Assurance (QA) professional—to determine if the proposed design meets the initial set of business goals. Testing may be repeated, specifically to check for errors, bugs, and interoperability. This testing will be performed until the end-user finds it acceptable. Another part of this phase is verification and validation, both of which will help ensure the program"s successful completion.

6. **Implementation**

   The sixth phase is when the majority of the code for the program is written. Additionally, this phase involves the actual installation of the newly-developed system. This step puts the project into production by moving the data and components from the old system and placing them in the new system via a direct cutover. While this can be a risky (and complicated) move, the cutover typically happens during off-peak hours, thus minimizing the risk. Both system analysts and end-users should now see the realization of the project that has implemented changes.

7. **Operations and Maintenance**

   The seventh and final phase involves maintenance and regular required updates. This step is when end users can fine-tune the system, if they wish, to boost performance, add new capabilities or meet additional user requirements.