**1. What is V model? Advantage and disadvantage? (Hoàng Hường)**

The V-Model (Validation and Verification Model) is a software development model that emphasizes the importance of verification and validation at every phase of the software development lifecycle. It's an extension of the waterfall model and is known for its systematic and sequential approach.

**Advantage:**

**Simple and Easy to Use**: The model is straightforward and easy to understand.

**Verification and Validation at Every Stage**: Helps in catching defects early in the lifecycle.

**Structured Approach**: Provides a disciplined approach, with well-defined stages and deliverables.

**Early Test Planning**: Test planning starts early in the development phase.

**Ease of Maintenance**: Easier to manage changes and maintain the system.

**Disadvantage**

**Inflexibility**: Like the waterfall model, the V-Model is not flexible and changes can be costly.

**Difficulty in Handling Changes**: Requirements must be very clear before the project starts, making it less ideal for projects with uncertain or evolving requirements.

**No Early Prototypes**: The model does not produce early prototypes, which can be a drawback for projects where user interfaces are crucial.

**Higher Risk and Uncertainty**: The model is less suitable for complex and object-oriented projects because of its linear nature.

**Sequential Execution**: The V-Model assumes that every phase must be completed before the next phase begins, which is not always practical.

**2. What is waterfall model? Advantage and disadvantage? (Hoàng Hường)**

The Waterfall Model is one of the earliest models used in software development. It is a linear and sequential approach where each phase must be completed before the next phase begins. The model is often visualized as cascading steps, similar to a waterfall.

**Advantage:**

**Simple and Easy to Understand**: Its linear approach makes it straightforward to follow.

**Well-Defined Phases**: Each phase has specific deliverables and a review process.

**Easy to Manage**: Due to its rigidity, it is easy to manage, with each phase having clear milestones.

**Disadvantage**

**Inflexibility**: Once a phase is completed, going back to make changes is difficult and costly.

**Late Testing**: Testing is done after implementation, making it hard to identify issues early.

**Poor Handling of Uncertainty**: It assumes that requirements are well understood from the start, which is often not the case in real-world projects.

**High Risk and Uncertainty**: The model does not accommodate changes well, leading to high risk and uncertainty, especially for long-term projects.

**3. What is Agile model? Advantage and disadvantage?(Tùng Anh)**

The Agile model is a software development approach that delivers work in small, iterative increments, usually in 2-4 week sprints.

Advantages:

Flexibility: Easily adapt to changes.

Customer Collaboration: Frequent feedback ensures the product meets customer needs.

Faster Delivery: Regular releases of working software.

Improved Quality: Continuous testing and integration.

Disadvantages:

Scope Creep: Uncontrolled changes can expand the project scope.

Requires Experienced Team: High level of collaboration needed.

Lack of Documentation: Focuses on working software over detailed documentation.

Less Predictable: Harder to estimate time and cost.

**4. Why do we need to test on multiple browsers? Why do we need to test on multiple version? (Tùng Anh)**

Testing on multiple browsers:

Different Rendering: Browsers display pages differently.

Browser-Specific Bugs: Some bugs only appear in certain browsers.

User Preferences: Ensures a consistent experience for all users.

Testing on multiple versions:

Backward Compatibility: Ensures older versions work well.

Security and Stability: Identifies version-specific issues.

Feature Support: Ensures functionality even if newer features are unavailable

**5. How to choose the browser and version of browser to execute test? (Bảo Trân)**

Analyze User Data:

* User Analytics: Use tools like Google Analytics to gather data on which browsers and versions your users are primarily using.
* Market Research: Look at market share reports from sources like StatCounter or W3Counter to understand the most commonly used browsers and versions globally or in your target market.

Consider Project Requirements:

* Client Requirements: Some projects will have specific browser requirements dictated by the client.
* Application Type: For example, if your application is heavily used in enterprise environments, you may need to focus on browsers like Internet Explorer or older versions of Edge.

Include Popular Browsers:

* Mainstream Browsers: Ensure you test on the most widely used browsers, typically including Chrome, Firefox, Safari, Edge, and sometimes Opera.-
* Mobile Browsers: If your application is mobile-responsive, include mobile browsers like Chrome for Android and Safari for iOS.

Test Different Versions:

* Latest Version: Always test on the latest version of each browser to ensure compatibility with the newest features and security updates.
* Older Versions: Depending on your user base, test a few older versions to ensure backward compatibility, especially for browsers with significant market share in older versions (e.g., Internet Explorer 11).
* Beta/Developer Versions: Occasionally test on beta or developer versions of browsers to anticipate and prepare for upcoming changes.

**6. What if customer find a bug on a browsers that are not listed on required browser? (Bảo Trân)**

* Verify the Bug: Check if the bug is reproducible on the reported browser. Ensure it is not due to browser-specific extensions or configurations.
* Document the Bug: Record detailed information about the bug, including a description, screenshots, and steps to reproduce it.
* Compatibility Check: Evaluate your application's compatibility with other browsers. Even if the requirements specify certain browsers, it is good practice to ensure broader compatibility with popular browsers.
* Assess Impact: Determine the severity and impact of the bug on users. If the bug significantly affects user experience, consider prioritizing its fix even if the browser is not officially supported.
* Update Requirements: Consider expanding the list of supported browsers if a significant number of users are using the browser where the bug was found.
* Communicate with the Customer: Inform the customer that you have received their bug report and are investigating it. Provide an estimated timeline for resolution if possible.