

Explain 2 example issues each for different phases of app development discussed: project planning, user interface (UI) design, programming languages specific to the target platform (like Java for Android or Swift for iOS), database management, application logic, testing methodologies, deployment processes, and maintenance strategies; essentially, everything needed to conceptualize, design, build, test, and launch a fully functional mobile application across different operating systems.

1. Project Planning

Inadequate communication: Inadequate communication between team members can lead to misunderstandings, decreased team morale, and project delays

Inaccurate Time Estimation: Underestimating the time required for development phases, resulting in rushed implementation or missed deadlines.

2. User Interface (UI) Design

Inconsistent Design Elements: Lack of standardized design principles can lead to mismatched fonts, colors, or button styles, confusing the user.

Poor Accessibility: Neglecting accessibility guidelines (e.g., screen reader support) may alienate users with disabilities.

3. Programming Languages Specific to Target Platforms Limited Cross-Platform Compatibility

Choosing a platform-specific language might complicate cross-platform development and require separate codebases.

Performance Bottlenecks: Developers unfamiliar with the chosen language may write inefficient code, leading to app lag or crashes.

4. Database Management Inefficient Query Design: Poorly optimized database queries can slow down data retrieval, negatively impacting app performance.

Data Sync Issues: In mobile applications, keeping data synchronized between local storage and the cloud can be challenging and if not done properly, can cause issues.

5. Application Logic

Poor Code Organization: Spaghetti code or a lack of modularization can make the app difficult to debug, extend, or maintain.

Hardcoding Values: Relying on hardcoded values instead of dynamic configurations can limit scalability and adaptability.

6. Testing Methodologies

Inadequate Coverage: Focusing only on functional testing and neglecting edge cases can leave bugs undiscovered.

Device Fragmentation: Testing on limited devices may overlook issues on less common operating systems, screen sizes, or hardware configurations.

7. Deployment Processes

App Store Guidelines Non-Compliance: Failing to adhere to guidelines for platforms like the Apple App Store or Google Play can result in rejection during the review process.

Version Control Errors: Mismanagement of version updates (e.g., deploying the wrong build) can lead to incomplete or broken app features.

8. Maintenance Strategies Lack of Bug Fix Priority: Ignoring reported bugs or delaying updates can damage user trust and retention.

Dependency Updates: Failing to update libraries or frameworks can leave the app vulnerable to security risks or compatibility issues.

