Requirement Engineering in Mobile Development

Presenter
Gagandeep Garg
Outline

➢ Introduction and Role of RE
➢ Challenges
➢ Distinct Behaviour
➢ Mobile Design First
➢ Methods
➢ Conclusion
Mobile Development:

➢ It is an act or process in which application software is developed for mobile devices

➢ Platforms Required:

1. **Front end development tools**: Focused on the user interface and user experience. Examples: Android sdk, iOS sdk

2. **Back end servers**: Provide a set of reusable services that are centrally managed and controlled. Example: IBM MobileFirst Server (plugin for Eclipse)

3. **Testing**: Mobile applications are tested within the development environment using emulators. Example: Google Android Emulators.
Requirements for Mobile Development

➢ User requirements are changing with time and this change happens frequently

➢ Development time is short

➢ Emphasis is more on user interfaces
Role of Requirements Engineer

➢ Verify the behaviour of application in different Networks (3G, 4G, LTE, Wi Fi)

➢ Techniques needed for assuring data integrity

➢ Trade off between user interface and non functional requirements

➢ How to maximizes the battery life and resource usage
Importance

➢ Manage Complex Requirements
➢ Improve Team Collaboration
➢ High Quality Products
  ○ Consistency
  ○ Completeness
  ○ Correctness
  ○ Unambiguous
  ○ Maintainability
Challenges

1. Requirement Engineering is not integrated to development method

Development Method = problem domain + structure of product base + process + organisation

RE provides the framework for problems domains, The method which weakly supports RE is destined to fail.

RE is not integrated in existing product and methods
Challenges

2. RE process is not collaborative

➢ Wrong Time - Formalisation of requirement is needed

➢ Wrong People - Little contribution of developers in early phases

➢ Wrong Techniques - Marketing experts lacks technical skills.

➢ Wrong Process - Results in complex Architecture
Challenges

3. Moving toward fragmentation rather than unification

➢ Fragmentation across platforms
  ○ Different UI, UX, HCI standards, programming languages and tools
  ○ Examples: Android uses Java, iOS uses Swift language

➢ Fragmentation within same platform
  ○ Every device has its own properties - memory, CPU speed
  ○ Example: Android devices with different screen sizes and resolution
What makes Mobile Different ??

According to Wikipedia.com

“An embedded application is a computer system with a dedicated function within a larger mechanical or electrical system, often with real-time computing constraints.”
What makes Mobile Different ??

• Potential interaction with other applications - embedded devices normally have pre-installed software.

• Sensor handling - Accelerometer responds to device movement

• User interfaces - Adhere to guidelines set forth by platform provider.

• Power consumption - Extensive use of battery draining resources
What makes Mobile Different??

- Complexity of testing - Added complexity of networks and transmission through gateways
- Security - open allowing the installation of malware
- Families of hardware and software platforms – multiversion support
Mobile First Design

➢ Why Mobile First??

○ Flip of workflow – Mobile consideration first than desktop version based on statistics.

○ Large User base – In the U.S., 25% of mobile Web users are mobile only.

➢ Graceful degradation vs progressive Enhancement

○ Graceful degradation method have to design functions on as many browsers and platforms as possible.

○ Progressive enhancement method helps the Requirements Engineer to focus their energy on one platform, rather than multiple platforms.
Mobile First Design

➢ Why progressive enhancement wins
   ○ Desktop version first - Take advantage of everything, none scales to mobile application
   ○ Mobile version first - Leaner model, quicker to develop, focus on new features in desktop

➢ Downsides to Mobile first design
   ○ Constraints - Small screen size, fewer resources, battery, platform updates
   ○ Immature - New field

➢ RE has the opportunity to really meet needs of the user in a smaller and focused list rather than a larger system.
Methods

1. RELAX Method

2. Proposed Continuous RE

3. Proposed Agile Scrum Method
RELAX

Mobile systems need to dynamically self-adapt to changing environmental conditions to provide reduced functionality.

Changing environmental conditions such as sensor failures, malicious threats, and unexpected input.

RELAX process documents what environmental changes can affect the requirements and how they can be partially satisfied.
RELAX

For each SHALL statement DO the following:

1. Must SHALL stmt always be satisfied?
   - YES: INVARIANT requirement
   - NO: 2. Identify Uncertainty factors

2. Identify Uncertainty factors
   - YES: 4. Introduce RELAX operator(s)
   - NO: 3. Must SHALL statement be RELAX-ed to handle Uncertainty factors?

3. Must SHALL statement be RELAX-ed to handle Uncertainty factors?
   - YES: RELAXed requirement
   - NO: 1. Must SHALL stmt always be satisfied?

Figure 2. RELAX Process
Proposed Continuous RE

This technique separate the concerns

- **Generic Space** - Where new models are developed with reference from already existing models (Epics).
- **Conceptual space** - Multiple Technical Space - Each serve some specific concerns
- **Functional space** - Kind of Technical space Serves the functional and design representation
Proposed Continuous RE

Each Stakeholder in RE has different viewpoint

The development process starts with:

1. The objective of mobile application
2. Stakeholder Analysis
3. Create a vision.
4. Role Distribution
5. Then RE process starts

In Generic space first Epics are created.
Epics breakage:
1. User interface components
2. Phases
Proposed Continuous RE

1. Gather information sources.
2. Organize information sources.
3. Elicit requirements.
4. Analyse and indicate requirements.
5. Map requirements to specific elements.
6. Split requirements.
7. Define goals for new elements from requirements.
8. Define goals to requirements.
9. Integrate the requirement.
10. Validate requirements.
Proposed Agile-Scrum Method

Phase 1: Requirements Analysis.
Phase 2: Design and Development
Phase 3: Test and QA
Phase 4: Product acceptance phase
Phase 5: Release to Market.

uses sub-version repository to control and maintain versioning of all project (Git, Bitbucket)
Conclusion

➢ Mobile Industry is always changing and involving new features

➢ Approach of RE should be adaptive like RELAX

➢ Focus on mobile first design paradigm

➢ In the end there is no CATCHALL method for RE and planning in mobile development
Thanks and Discussions !!!