Scenario

Contributors:

Ethan Nelson (Team lead)

Suzy Naegle (Scrum Master)

Matthew Tyra (Functional manager)

Samuel Despain (Chief Editor)

Atsushi Jindo (Quality Assurance)

Overview of Elicitation Requirement:

There are times where the sponsor may have difficulties expressing the needs of his software. However while the solution may elude the sponsor, it always is easier to focus on the problems. Focusing on the problems is a way to gain an insight about what the sponsor wants. This is where the scenario requirement comes in. Scenarios work by giving the sponsor different scenarios and documenting their response to them. There are three different types of scenarios: [1]

* Problem Scenarios – Describe problems encountered in performing their work.
* Activity Scenarios – Describe how users perform their daily work.
* Interaction Scenarios – Describe how users interact with the system.

When documenting the response to scenarios you can classify the answer into two parts: [1]

* Pain Relieves
* Gain Creators

Pain Relieves are how the problem is solved. This helps the development process be less strenuous and frustrating. [1]

Gain creators help the user do more with less by making things simpler in the program or even increase potential sales. [1]

The Elicitation technique Scenario, also known a Use Cases, lays out the possible user interactions with the product to achieve specific goals, showing all the possible interactions and exceptions. This technique can be very good to find flaws in the design and help simplify the product.

“Use cases add value because they help explain how the system should behave and, in the process, they also help brainstorm what could go wrong.” [3] The cons are it does not include: Implementation-specific language or Details about the user interfaces or screens. [3]

To implement this technique, we could create a user case for a business posting an activity or deal, one for a user creating an activity, one for a user joining an activity, etc.

The Scenario elicitation technique involves painting a story of how your product will operate in a specific typical instance. This allows the clients to see their potential product in an instance and how the product will react. This technique creates a better form of communication between client and developer. Active story telling can better transfer information and is invaluable in instances where a large amount of the product involves user interface. [4]

* Involves painting a story of how your product will work
* Used to satisfy uneasy clients
* Provides detailed description of a specific instance of your system operating
* Tells of a specific typical instance

Ubiquitous computing aims to enhance computer use by not only making many computers available through the physical environment but making them effectively invisible to the user. **Scenarios** are examples of interaction sessions and consist of descriptions of sequential actions which relate to real-life examples rather than abstract descriptions of the functions. [2]

Among the reasons to create test scenarios include ensuring complete test coverage, stakeholder approval to make sure the product is tested thoroughly, being able to quickly form a proposal for clients, determining the real use of the application, and studying the essential end-to-end transactions. However, test scenarios should not be used when the application is overly complicated or unstable, follows Agile methodologies, or for new bug fixes or regression testing. [5]

Technique plan:

* Each team member to come up with two different scenarios or problems that could potential happen within the application.
* Present 5 scenarios to team on Friday and collectively come up with solutions to the different scenarios.
* Present 5 scenarios to other teams on Friday and see their collective answers
* Record scenario data

Raw data:

Problem 1: Local businesses getting hidden by big businesses who can afford to pay the most.

Problem 2: People posting a million activities and spamming the activity list.

Problem 3: An overwhelming number of initial users signing up at once, possibly causing server crashes.

Problem 4: Users’ personal information leakage

Problem 5: A user who creates an event must cancel it. Does someone else become the leader or does the whole event get cancelled?

Problem 6: There could be some verification problems down the road. While Rexburg is a safe place it is kind of sketchy to meet with strangers and there could be phishing done through the app. False events/activities.

Correlated data:

* Local businesses getting hidden by big businesses who can afford to pay the most.
  + Users can filter activities by location, price, and by local vs chain businesses.
  + Different bidding tiers for local and chain businesses
  + Limited paid spots
* People posting a million activities and spamming the activity list.
  + Limit the number of activities a user can post: limit per day/ per activity
  + Reporting activities option to limit spam
  + Recurring activities
* An overwhelming number of initial users signing up at once, possibly causing server crashes.
  + Do not skimp on servers – a backup server might be helpful.
  + Beta testing to test/predict growth and slowly allow people to join by word of mouth
  + Locally stored data
* Users’ personal information leakage
  + Encrypt personal information on all user accounts
  + Locally stored data
* A user who creates an event must cancel it. Does someone else become the leader or does the whole event get cancelled?
  + Everyone gets cancellation notification, someone else has the opportunity to become the leader. If no group leader is picked within a time limit or everyone rejects the notification, then the event is canceled
* There could be some verification problems down the road. While Rexburg is a safe place it is kind of sketchy to meet with strangers and there could be phishing done through the app. False events/activities.
  + Personal rating system to rank others as ‘good’ or ‘bad’ users. Only certain ranks can go on 2 person activities
  + Company has access to data on app. Activity history is available. GPS location available through the app
  + Emergency contact or report user options

Conclusion:

It is important to look into all aspect of software while developing it. It is important to look for different scenarios that a user could encounter.

Research Citation:

[1] J. Parker, “Personas and Scenarios as a Requirements Elicitation Technique,” *Enfocus Solutions Inc*, 12-Oct-2014. [Online]. Available: https://enfocussolutions.com/personas-and-scenarios-as-a-requirements-elicitation-technique/. [Accessed: 05-May-2020].

[2] Alawairdhi, M., & Aleisa, E. (2011). A Scenario-Based Approach for Requirements Elicitation for Software Systems Complying with the Utilization of Ubiquitous Computing Technologies. *2011 IEEE 35th Annual Computer Software and Applications Conference Workshops, Computer Software and Applications Conference Workshops (COMPSACW), 2011 IEEE 35th Annual*, 341–344. <https://doi.org/10.1109/COMPSACW.2011.63>

[3] A. S. for P. Affairs, “Use Cases,” Usability.gov, 09-Oct-2013. [Online]. Available: https://www.usability.gov/how-to-and-tools/methods/use-cases.html. [Accessed: 06-May-2020].

[4] P. Loucopoulos & V. Karakostas, Chapter 3 - Requirements Elicitation, 01-Sept-2008. [Online]. Available: <https://personal.utdallas.edu/~chung/RE/RE_chapters/Chapter%203.pdf>

[5] What is Test Scenario? Template with Examples. (n.d.). Retrieved May 06, 2020, from <https://www.guru99.com/test-scenario.html>