Requirements Gathering

CS-370: Software Design and Development

Robert Bruce

What is requirements gathering?

- Requirements gathering is one of the first phases of application development.
- Requirements gathering is the process of identifying functionality and features which are necessary to implement an application.
- Well-defined requirements are clear, concise, consistent, unambiguous, and verifiable.
- Words to avoid in requirements gathering include*:
  - Comparatives such as “faster”, “better”, “shinier”, or “more”.
  - Imprecise adjectives such as “fast”, “robust”, “user-friendly”, “efficient”, “flexible”, or “glorius”.
  - Vague commands such as “minimize”, “maximize”, “improve”, and “optimize”.

* Source: page 89, *Beginning Software Engineering* (2nd edition) by Rod Stephens

What happens after requirements gathering?

- Once the requirements gathering is completed (which could be a long list), the next step is *prioritization* of those requirements:
  - The MOSCOW method can be used for prioritizing requirements (must include, should include, could include, and will not include).
- Once the requirements are prioritized, the final step is to define the prototype of the application as a Minimum Viable Product (MVP).

Types of requirements gathering

- *Business requirements*: define the high-level goals of an application; specifically, an explanation on what the customer wishes to achieve. Business requirements are used for marketing targets.
- *User requirements* (also called *Stakeholder requirements*): define how the application will be used by end-users.
- *Functional requirements*: describe capabilities of the application. Note: similar to user requirements (above) but could include system capabilities not visible to the user.
- *Non-functional requirements*: define measurable ways to determine the extent an application meets desired goals or qualities. Examples include benchmark tests for performance, reliability, or security.
- *Implementation requirements*: define temporary transitional features which are necessary to complete the application. Example: procuring new computing hardware, hiring new staff, or preparing training materials.

Source: pages 90-92, *Beginning Software Engineering* (2nd edition) by Rod Stephens

What is FURPS?

- FURPS is a method for classifying requirements into five specific categories.
- FURPS is an acronym which stands for:
  - Functionality: What the application will do.
  - Usability: What the program should look like.
  - Reliability: How reliable the system should be.
  - Performance: How effective the system should be.
  - Supportability: How easy is it to support the application.

Source: pages 92-93, *Beginning Software Engineering* (2nd edition) by Rod Stephens

How does one gather requirements?

- **Listen to the customer!**
- Interview the customer by asking the following questions:
  - *Who* will be using the application?
  - *What* does the application need to do?
  - *When* does the application need to be completed?
  - *Where* will the application be used (e.g. desktop app, mobile app, tablet app, etc.)?
  - *Why* does the customer need this application (i.e. clarify customer's needs)?
  - *How* should the application be implemented? Sometimes customers have insightful ideas on how the application should work!
- Study and observe user work-flow. Studying the user workflow (the pipeline) may reveal requirements the customer didn't realize they needed!

Source: pages 98-99, *Beginning Software Engineering* (2nd edition) by Rod Stephens