

Comments to classmates

Have you started your Work Diary? (Assignment 1 – page 2)

Please check your uowmail inbox daily!

Suggestions for interviewing the customer

Roleplay! Pretend that this is a real business meeting. Greet the customer warmly, smile and be polite. Stay standing until they say or indicate that you should sit down.

Make small-talk so that people relax.

Ask about their business. What kind of business do they do? How old is the company? Public or privately owned? How many people in the company? How many offices in Australia or across the world? Try to get an overview of what they do, and how they do it.

Take notes! Write clearly so that you can understand your notes later!

Bear in mind while interviewing the customer that because this is a uni assignment, the staff don't want to spend a lot of time with each group. The answers to many questions may be "As you like". In a business environment, that's not acceptable, but in a uni assignment it's fine ☺ We can make stuff up.

Remember to get good detail on their **existing** process for bugs. What information do they record? Try to make a flowchart for their process. We need to make our product suit their process.

They may also ask "Well, what do you think/suggest?" At this point, refer to the "Suggested Features" list that we prepared this week. Don't give the whole list! Just talk about the ones relevant to the question you asked. The more suggestions you make, the more work we have to do! ☺

For each requested feature, ask if it's Essential, Desirable or Optional. This will help us prioritise our work.

After all the questions have been asked and answered, ask one final question. "Is there anything else that comes to mind that we should know about?"

When closing the meeting, thank the customer for their time and the opportunity to work with them. Tell them that we'll be in touch if we have any more questions, otherwise we'll give them our estimate soon.

Bug Tracker – Scoping Questions

System Overview and Scaling

- Tell me about your IT environment. (Desktop platforms, networking infrastructure, office locations, etc.)
- Tell me about your IT team. (Where are they? One office or many?)
- Tell me about your users. (One office connected by LAN, distributed across a WAN)
- How will you be using the system? What kind of bugs will the system be tracking?
 - Computer software development? If so, what languages will it need to support?
- How will this system integrate with your existing infrastructure?
- How many bugs each week/month do you anticipate?
- How many *concurrent* users you expect to be using the system?
- Do you expect this to grow? If so, by how much over what time frame?
- What's the desired completion date?

Bug Report Details

- Please describe the life-cycle of a bug in your organisation. (Make notes of their current process. We need to fit the program to their process, not the other way around.)
- What fields you need when raising a new bug?
- Which fields are mandatory when creating a new bug?
- How many statuses that we have for a bug? What are they?
- How many priorities for a bug? What are they?
- Work to develop a Risk matrix for bugs. How do they prioritise bugs? Impact? Age? Etc.

System Details

- What are the key features that you expect the system to provide?
- What operator roles will you need? E.g. Reporter, Developer, Triage, Supervisor, Manager, Reviewer, etc.
- What role-specific features have you determined? (e.g. Supervisor can edit any bug, Developer can edit only assigned bugs, Reporter can edit only his bugs)
- What criteria should we search when searching a bug? Just the subject? The description? Comments?
- What information should user records store?
- How would you like to implement Reputation?
- Should we restrict editing to the assigned developer? This will help avoid concurrent edit conflicts without complex record locking.

User Interface Details

- Look & Feel – is there a desired cultural aesthetic preference? E.g. A system targeting users of a single culture may have some different traits
- What (features/buttons) will you need on the UI of every user?
- What operator languages must the system support?
 - Asian character sets? Right-to-left text?

Bug Report – Suggested Fields

- Date of creation
- Time of creation
- Author
- Short description
- Long description
- Priority
- Impact
- Owner (the assigned developer)
- Lifecycle / Status (These will need to be set to mimic the customer's current process)
 - New (Submitted by reporter)
 - Prioritised (Triage operator has reviewed and prioritised but not assigned the bug)
 - Assigned (Bug has been assigned to a Dev)
 - Accepted (Dev has accepted the bug onto his/her work queue)
 - Open (Dev is actively working on the bug)
 - UAT (User Acceptance Testing – fixes for complex bugs may require live testing before they can be deemed successful fixes)
 - Resolved (Dev has resolved the issue and submitted a patch or hotfix)
 - Closed (Bug report has been reviewed and accepted as completed)
- Resolution
- Links to patches
- Links to related bug reports
- Links to knowledgebase articles
- Reviewer

Bug Tracker – Feature Suggestions

First Phase

- Ability to log user in/out
- Ability to report a new bug
- Ability to edit a bug report
- Ability to close a bug report
- Overview / Browse bugs screen
- Keyword search of existing bugs
- Support for multiple roles with different access levels
 - Triage, Developer, Reporter, Manager, ...?
- Ability to register a new user
 - Self-registration or only by authorised roles? – SR
- Ability to manage user detail / profile
-

Second Phase

- Reporter reputation tracking
- Developer reputation tracking
- Ability to mark a report as a duplicate / invalid
- Categorisation scheme
- Ability to generate reports
- Ability for users to add new categories (which roles have this access?)
- Ability to add a comment to a bug report
 - Available to all, or just assigned developer? – SR

Features for a Real System that we Don't Want in this Project

- Spell-checker in search box
- Regex support for searching
- Distributed access (servers in each office that need to synchronise with head office)
- System to provide suggestions of possible solutions
 - Difficult to implement. Leave out of scope unless customer asks for it. - SR
- Ability to share bugs on web to get additional help
 - Outside of scope unless the customer asks for it. - SR
- Knowledgebase
- Multi-language support
 - They will say that it's out of scope for this assignment - SR