Name: Interviewee

	Timespan	Content
1	0:00.0 - 0:13.3	How do customers request modifications or report problems?
2	0:13.3 - 1:59.6	Customers report problems either through e-mail or through the customer's JIRA. The problem may be a defect (i.e. something that went wrong in the system) or a change request. On one hand, when customers start using JIRA we tell them to take care while reporting change requests, because it means that they need a new behaviour in the application, something that was not in the requirements specification. Customers charged extra fees for change requests as they are out of the initial scope of the project. On the other hand, defects are bugs in the system that we will handle any way. In the JIRA tool, the customers interface is a form where he provides a summary and a description for the problem. They can also attach screen shots, or data files for example and excel sheet that they were trying to import and failed. They also link the defect to one of the requirements (i.e. use cases) that are already written. This helps us analyse the use and revise its scenario to evaluate whether this is a valid defect or not. Sometimes customers report problems orally through telephone. The only two written ways are e-mail or JIRA. Feedbacks reported through e-mail are unstructured and unorganized.
3	1:59.6 - 2:07.6	Do you think when customers use JIRA it helps them provide better descriptions? (i.e. does the tool assure this)
4	2:07.6 - 2:34.2	The JIRA tool allows for the above mentioned details to be entered however it does not assure that the customer enters a minimum amount of valid information. Sometimes they provide enough information while in other times they give brief information that is not enough to interpret the problem, so we have to ask them to give more details. I think it is training process, customers at first are not used to providing their feedback or reporting problems. But by time they get used to it and they know the essence of giving more details, so they start improving the feedbacks they provide.
5	2:34.2 - 2:42.2	Is the customer able to differentiate in his feedbacks whether it is a defect or it is change request?
6	2:42.2 - 3:16.8	No. Most customers in the beginning report every problem as a defect and needs to be fixed. In the JIRA the issue goes through a workflow. One of the states in the workflow is whether the issue is a qualified defect or not. 80 % of the problems are reported as defects at first. It is the consultant's job to qualify whether it is a valid defect or not.
7	3:16.8 - 3:22.1	What are the criteria upon which the consultant qualifies whether a defect is valid or not?
8	3:22.1 - 4:04.6	Based on the requirements document that was agreed upon in the beginning of the project. Therefore if the customer relates the problem to a feature in the requirements document then it's a defect. If not then it is a change request. Also, change requests can relate to a feature in a requirements document but with a different behaviour.
9	4:04.6 - 4:47.2	What are the problems that occur during the perception of customer requests? How do you handle these problems?
10	4:47.2 - 5:51.0	As a consultant first thing I do is try to reproduce the scenario/ defect described in the problem internally on our test system. If I successfully reproduced the defect therefore the customer has supplied a correct description, if not then the given scenario is incorrect or it is not a defect. Second, I check the requirements specification document to see what was written regarding the feature. Examples of problems that can occur is that users do not write a complete scenario (steps) to the problem so I do not understand how did the problem occur to him. Also, lack of screen shots is another important problem in customers' feedbacks.

11	5:51.0 - 6:01.6	Are the problems communicated with the customer in the perception and reproduction phase?
12	6:01.6 - 6:28.2	Mainly, we communicate with customers when there is a problem or after finishing the whole qualification phase. Problems that may occur like the written scenario is invalid, or the customer does not understand the feature correctly. So we give him feedback with the corrections (e.g. missing steps), or asking for extra information.
13	6:28.2 - 6:36.2	Do you use any tools to help you in the analysis or interpretation of feedbacks?
14	6:36.2 - 6:49.5	No we do not use any specific purpose tools. But the application itself is a reference that aids this task.
15	6:49.5 - 7:37.3	What are the problems that occur in the estimation tasks? What kind of information from the users that can inform your task estimation decisions?
16	7:37.3 - 9:18.4	If the customer gave a good description for the problem, or after asking him for more information we reached a good description, we transform this information into a specification of what exactly are we going to do to resolve this problem. This written specification is sent to the customer. Based on this written specification the estimation is done. Customers are not usually involved in the estimation tasks. However, in some projects they might get involved in that phase. They can get involved after the estimation is handed to them, especially when it is a big effort or a change that was not included in the initial project. So we send it as if we are sending a new offer including the scenario and estimation (time and cost) and wait for their response. If the problem is a defect we make internal estimation and start working on it immediately. Projects managers can better give examples and details of problems that occur during the estimation.
17	9:18.4 - 9:55.6	What are the approaches you use in order to identify the impact of a change on the system?
18	9:55.6 - 12:51.1	The customers are always not aware of the impact of the changes they request on the system. Especially because the system is used by multiple users and each user uses a specific module in their daily work so they are not fully aware of all the functionalities in the system. For example, if the user wants to change a certain equation in the module he uses instead of adding, he want it to multiply. Mostly, he is unaware that another user will be affected by this change. Two main roles are responsible for examining the impact of a change. The consultant who is aware of the impact of changes because he knows the requirements and the relationships between them. Also, the developer who is aware through the implementation of the modules that will be affected from a change, because he knows where a function could be usable from several places in the code. The developers examines the change and reports the affected places, consultants get back to the customer and reports the affected parts and tells them to discuss and check with their colleagues. The customer is made aware of the impact and its cost and he either resolves this issue internally with his colleagues or we try to meet half way.
19	12:51.1 - 13:52.2	What are the customer-to-engineer communication problems that affect your design decisions and the way you approach the problem?
20	13:52.2 - 15:25.3	Factors that affect design decisions are: Priority and severeness. For example if a problem is a blocker and the customer needs it maximum tomorrow. Therefore we will do a workaround until a clean solution can be made. Also, the budget is an important factor whether the customer has no budget or not willing to pay for a clean solution, or we have no budget on the project to make a clean solution. Out of honesty we inform the customer that there is a clean solution, but due to budget issues we will make him the workaround. Moreover, Quality is an important factor; if we have to consider quality then we have to go for a clean solution. If we will not consider the quality

		because it is an unimportant feature that is used annually then we can make a workaround.
21	15:25.3 - 15:43.9	From your observations what are the criteria that customers put into consideration while prioritizing their problems?
22	15:43.9 - 17:51.5	On the JIRA we have types for the priorities. First, it can be a UI issue like a spelling mistake or a grammar mistake or a color that the user wants to change. UI issue take "low priority" as it is not a blocking issue (i.e. it is not blocking the flow of the user's task). Second if a problem is totally blocking the user's application for example the application went down, then it takes a "high priority". There other intermediate levels of priority like "critical", where a certain scenario in the application is not working. Also, "Major" is a level of priority where there is a problem but not interrupting the flow of work and therefore it is not urgent to be done right away. The definitions of the priority levels are written on JIRA, and in the customer training process we tell them to enter the problems with the correct priorities. Example, they should not report a problem as critical while it is not. Also, as part of our process we qualify whether this priority is correct or not. If there is a conflict in opinions, for example, a problem the customer categorized it as a blocker and the consultant does not have the same opinion, then we discuss with him and explain our opinion. We inform him that it will take less priority so it will be put in queue and will be resolved later.
23	17:51.5 - 18:20.8	What are the types of conflicts that you encounter while negotiating with customers on priorities?
24	18:20.8 - 20:55.0	In some cases there are some problems that are high in priority but not urgent. For example if there is a scenario in the application that has many problems, but the customer will use it next month. Therefore, we consider this as a high priority because it is not working at all but not urgent. And the other way around that there are some problems that are urgent to the customer although they are not high in priority. Example, a sentence occurs incorrectly in a report, and the customer will hand in the report tomorrow for an official government organization, and it has to occur correctly. This is not a whole blocked scenario in the application but still it is urgent to the customer. This can be resolved by a workaround. We usually negotiate using arguments, workarounds, or trade-offs. Examples of trade-offs if the customer reported five issues we start by prioritizing them and discussing what is more important and what can be resolved later on. Or we give the user constraints like we only have time for resolving one issue per day, so give us a list of prioritized issue depending on that.
25	20:55.0 - 21:16.3	Can customers be mistaken in their usage of a certain feature?
26	21:16.3 - 23:10.6	Yes sometimes customers do not know how to use the application so they do things wrong, and they report problems as bugs, while actually it is lack of experience. Other times they report questions asking how certain tasks are done on the application, and users report it as a bug. This usually occurs in the beginning of their usage after we deliver the project, as users are still not experienced on doing tasks using the software. So they cannot judge whether it is the correct behaviour or it is a mistake in the application. Anyway they report it as a problem, and while we are in the qualification process of the problem we tell him that there is a misunderstanding and we explain the correct sequence for the behaviour. When we analyse the user's feedback if it was a question, we document a written response explaining a step-by-step scenario, the correct usage of the features, and what he was mistaken in. This documentation is very important so that customers do not claim that we did not respond or other users ask the same questions.
27	23:10.6 - 23:31.9	In your feedback system, do users see each other's feedback? Yes They do..
28	23:31.9 - 23:50.5	How can duplication happen when you have the same problem posted before with a response?
29	23:50.5 - 25:12.9	Most probably time has passed on the old feedback so it is not appearing in recent feedbacks. Or the new customer did not research old reported problems before he posted his problem to see if there are similar ones or not. Or if a user reported the problem in the past through e-mail then it

		will not be accessible to others afterwards. When we analyse that feedback, if the consultant realized that this issue is repeated he reports that it is a duplicate and then refer to the older issue number. The problem is if we do not remember we will go through the whole process again until it is resolved with this user. An automated recommendation in this part will save much effort on both the consultant and the user.
30	25:12.9 - 25:34.2	Did it happen before that you have suggested a solution to a problem and the user was not able to apply this solution?
31	25:34.2 - 26:14.1	For every issue there is a thread of comments. So when an issue is posted we reply with a solution, and it migh happen that the user cannot try the solution we gave. This is either lack of experience or the solution was not correct because we were not consider an aspect in their environment. When users report that they failed to try the solution we post other suggestions untill the issue is resolved.
32	26:14.1 - 26:32.7	Is there a way users can rate problems or vote for solutions?
33	26:32.7 - 27:04.6	No. It is unlikely that users vote they have the same problem or that a solution was beneficial to them. we may link a new issue to an older one by referencing the issue number for the old similar problem.
34	27:04.6 - 27:31.2	Do you have any comments, suggestions, or advice about our work that you would like to share?
35	27:31.2 - 30:18.7	Most important thing to me as a consultant is solving the duplication issue and linking feedbacks to each other. Using automated suggestions for customers is also important for example if the customer is posting a problem an automated message can occur "see issue number...", which contains solution to your problem. I think that voting is not important in our case, because our customers have limited number of users. Making use of historical information is a very good idea. For example, when a user is entering his feedback, depending on the keywords he used, a history of similar problems can occur to him. The structuring of the feedback is also important we need to have a variety of ways to describe feedbacks not only text. It is also important that users link their feedbacks (enhancements or defects) to a scenario. Also, feedbacks should be structured in way that makes sure that users entered valid misuse scenarios based on the data they used. Conflicts in prioritization are essential problems between us and the users. A more clear and refined definitions of priority levels and when to use them will be useful. Sometimes there is a misconception from the customers on what the application should do or behave like.
36	30:18.7 - 30:32.0	How do you resolve conflicts of misconception?
37	30:32.0 - 31:11.9	We try to persuade him based on our understanding to his business needs. Sometimes customers ask for features out of pretentiousness.
38	31:11.9 - 31:35.8	In release planning you divide the defects or change requests into smaller releases. What are the criteria that you base your prioritization upon? Do customers intervene in your decisions?
39	31:35.8 - 32:52.9	The project manager is in charge of deciding releases, and it is based on his understanding to the system's functionality. We try to release the work packages that will be most useful to the customer. We can implement parts from several modules but the customer will not be able to benefit form such release. So it's not by the number of modules we address in the release, but it should also be a whole working part as a whole. Also, release planning is influenced by the effort, because we estimate based on the available capacity. The customer is made aware of the releases and why we chose them in this order. Mostly, customers agree on our plan, because we consider all their business needs and any other aspects they gave. If they do not agree we discuss until we make a new release plan approved by both parties.

40	32:52.9 - 33:06.2	Any other final comments you would like to add?
41	33:06.2 - 36:46.9	There are problems that users report that are fixed indirectly. This might seem a a duplicate but it is not. This kind of problem is resolved within a previous fix of another related problem. So when the customer reports this need, we reply please see issue number... for similarity. One other subject related to requirements specification in the beginning of the project or even in the planning of a new enhancement, we write down the full description of the requirement and give it to the customer for review and reporting his opinion. This increases the understanding of the user to the system and whether it meets its expectations or not.