Case Learning

The Skills of Effective Communication in IT Project

As the development team leader, I work with about 30 people from different professions: developers, testers, administrators, designers, usability experts, project managers and people on the client's side. Working in such an environment, I've identified the main obstacles to effective communication.

I've also been involved in devising techniques to overcome those obstacles.

Email

Email is the primary means of internal and external communication at most companies. Surprisingly, many people still don't know how to use it properly.To make sure your emails are well received you need to make them clear, concise and actionable. In addition, using the appropriate format and knowing what to include or exclude from an email can help you maintain a professional presence.

The subject line is the first thing that a recipient notices. It should be brief and should explain the contents of the email. The recipient might want to refer to the correspondence in future, perhaps weeks or months later. Therefore, the subject line should clearly identify the project (including the customer, depending on the organization) and the subject matter.

Here are some examples of good subject lines:

- ACME Corp. | HR Portal | draft of functional documentation, v. 0.1
- ACME product page — questions after the meeting with marketing dept. on March 5th
- Please, send your report — deadline: March 10th

Here are some real-life examples of bad subject lines:

- ACME
- Question
- Request
- New images
- We're going for lunch at 1 pm

That last one is from a follow-up email that contained important documentation — true story!

Second, make it concise. People are busy, and they don't want to spend much time reading work emails. Make your emails as short as they can be, while still providing all the details necessary. Your sentences should be short and to-the-point. Before sending your email, read over it for any irrelevant information. If you have provided excess off-topic information, delete it. Some people try to send all their emails in under 5 sentences. An exercise like this will force you to focus only on the details that matter.

Another point is, write the email to be skimmed and acted on. Use formatting to help your reader get quickly to information they need. Bold the most important sentences in a long email. Include URLs or attachments if that will help the reader get the email processed faster. Never force the reader to hunt around for a URL or attachment in another email.

Good example URL practice:

"The agenda for the marketing meeting is in this Google doc URL. Google.com/sample"

Bad example URL practice:

"The agenda for the marketing meeting is in a previous email I sent you."

Issue-Tracking

Issue-tracking systems — including Redmine, Mantis BT, Bugzilla, Jira and many more — enable clients, developers and managers to work together in well-structured process.

Every company employs a workflow that makes sense for its business. Still, some rules apply to all situations, and here are ones that I've laid down after analyzing thousands of issues over the last few years:

- ✧ Title each issue as descriptively as possible. Remember that most people — whether developers, project managers or testers — deal with dozens of issues every day. They should get at least some sense of an issue's subject from its title. Thus, avoid titles like "Something's wrong." Rather, use self-explanatory titles, like "API throws NullPointerException when no attributes provided" Issues with such titles will be processed more quickly because they are easier to manage.

- ✧ Use attributes accordingly. Many trackers let you set special attributes on each issue, such as status, priority and category. Use them! The issues will be easier to sort, delegate and review.

- ✧ Write about only one thing per issue. For example, if you find many bugs in an application, treat every bug as a separate issue. This way, tasks can be assigned to different people, who can work on them simultaneously.

- ✧ Provide as much information as possible. Link to the buggy web page; write all of the steps needed to replicate an issue; attach screenshots. I worked with someone who attached a Word document with a few words and screenshots, titling the issue "Everything is in the attachment" — a big no-no. Write down in the comments section everything that happens with an issue. If someone explains something to you by phone or if a task is changed, document it.

- ✧ When working on a complicated project, I often find it useful to post important tasks on colorful sticky notes on a whiteboard (in addition to the issue-tracking system). Some are marked “waiting,” others “in progress,” yet others “done.” The color of the magnet affixed to each note indicates which team member is assigned to the task. Everybody can see the status of tasks and the progress of the project at a glance, which is the big advantage of this method.

Sticky notes on a whiteboard; such dashboards are a great place for daily stand-up meetings, too.

To sum up, keep everything in the issue tracker. Give each task its own issue. And describe the issue as best you can.