

Reducing TCO – best practices for integrating documentation with user interfaces

Jordan Stanchev

SAP Labs Bulgaria
June 2010

Agenda



- 1. General Rules & Processes**
- 2. Concrete Proposals**
- 3. Summary**

General Rules

- Write task oriented documentation.
- Do not describe self explanatory tasks.
- Avoid describing obvious tasks and repeating similar instructions across different procedures.
- Avoid publishing tutorials in your official documentation.
- Avoid describing system processes.

Processes

- Send direct feedback.
- Perform general reviews.
- Review dedicated text files.
- Plan the UI together with the responsible developer or development team prior to the actual development of the UI.

Agenda



- 1. General Rules & Processes**
- 2. Concrete Proposals**
- 3. Summary**

Tip 1: Perform general reviews



Send feedback in a way it could be easily consumed by developers. For example:

- in an attachment to an e-mail.
- directly in a file in your versioning system.
- as a simple Word file, containing a list of incorrect texts, and description how they should be corrected.

Note that the more precise you are in the feedback you give the easier it would be for your developers to find out what have to be improved in the source code they are developing. Such facilitation comes if you:

- cite the exact location of the UI you are reviewing in the software,
- refer to the exact text that you want to see improved,
- steps that you have followed to reach the UI in the software, and so on.

Tip 2: If you have an idea what the UI should look like, share it!



1. Imagine you are talking with a colleague of yours over a cup of coffee and he or she complains about the awful task that needs to be documented. Hearing someone is complaining about the complexity of the documentation that needs to be written is a sure sign that there is a UI that cries out for review!
2. Now imagine how you would want to have this documentation written. Would you want to have a simple wizard description? Or the opposite – would you want to have this described in a single task oriented procedure that talks only of using a single UI, instead of making you jump from one window on your screen to another?
3. Discuss your proposal for improvement with one or two of your colleagues to gather feedback, and propose it to the responsible developer!

Tip 3: Review error messages



Propose a set of rules for the error messages generated by your software. Talk to the developers and propose meaningful texts for the errors the software can make. Remember to put a proposal in the error message on what the customer should do once he or she faces that error.

Tip 4: Hint for new-comers



You have always to start from scratch. Once you become familiar with the software you have to describe in your documentation, you can easily forget to cover with documentation area that really requires special knowledge, that is documentation.

So focus on each point that you have problems understanding, while you are becoming more and more familiar with the software you will describe. Write these points down ASAP, and highlight them in the documentation.

Tip 5: Avoid describing fields that are self-explanatory



Using the UI as much as possible requires collaboration and cooperation with development. Some examples are:

- Influence/support development to keep the field labels self explanatory, and request they provide tooltips. Follow up to make sure this was done. Where tooltips are already available no further explanation is needed in the documentation.
- Provide text to be included by the developer at the top of a UI to explain what should be done in the screen. For example: "In this screen you can display the users that are assigned to the selected role."
- Describe only the necessary columns headings of a table from the UI. These headings are typically the search fields of the search screen.
- Document only those properties that are mandatory and not self-explanatory.
- Fields should be explained in the context of task-oriented procedure. Fields that are secondary to the procedure or are self-explanatory need not be explained.

Tip 6: Avoid inserting screenshots/diagrams in documentation that mirror the UI



Avoid inserting screenshots/diagrams in documentation that mirror the UI.

Such approach can be useful for you to get a quick overview of what you have to describe in your documentation. Rather, refer to/describe them in the context of the relevant procedures.

Often describing the UI look-and-feel does not bring any value to the customers, as it visualizes what they already see. Mirroring the UI does not bring much value, while it may take a lot of your time, and a lot of space in the documentation you create.

Agenda



- 1. General Rules & Processes**
- 2. Concrete Proposals**
- 3. Summary**

- Write task oriented documentation.
- Do not describe self explanatory tasks.
- Avoid describing obvious tasks and repeating similar instructions across different procedures.
- Avoid publishing tutorials in your official documentation.
- Avoid describing system processes.
- Perform general reviews
- If you have an idea what the UI should look like, share it!
- Review error messages
- Get advantage from not knowing details about the product.
- Avoid describing fields that are self-explanatory
- Avoid inserting screenshots/diagrams in documentation that mirror the UI

Thank you!

Questions?