Computer Science Capstone Design

Assignment: User Manual and Delivery to Client

(50 pts)

Overview

The project is done, you've put the last touch on the code, you're just finishing off the Final Report…woohoo! Time to relax, right? Not quite! You would be forgetting the most important step of the entire project, which is formally delivering the product to your client. This is much more than just popping your code on a thumb drive, taping it to a copy of the final report, and dropping it in your client's mailbox! In real life, your client is the one who pays you…and you won't be getting paid until you have formally delivered the product. This means meeting with the client, reviewing the final version of the product in a thoughtful step-by-step fashion to show how all key functionalities have been implemented, and then ensuring that the client *is able to effectively utilize your product*. This last step is key! It means that you product should be installed on whatever platform (web, their local office computer, whatever) the client specified, that the client has full access (e.g. root user login on that platform) to the software, and that you have delivered and shown the client all relevant documentation for operating your product.

The aim of formal product delivery is to make sure that the team has done its due diligence in installing the product to the client's satisfaction, and educating the client on how to operate the product. It consists of two parts: The product **User Manual**, and the **Product Delivery Check-off** Sheet.

The User Manual

The User Manual is perhaps the most important document that the team will produce from the client's perspective. Previous documents like the Requirements Document and the Design Document were primarily technical documents, really more for the team's use and for outside technical reviewers than for the client. The User Manual is the one document *written solely and specifically for the client*…and is also the document that the client will consult most going forward. A good user manual focuses on a single goal: educating the client on how to install, operate, and maintain the product.

You've all read million of "user manuals"; practically every piece of software you download online has a README and/or a "Getting Started" page on their website. You know exactly how goes: installation, configuration, management. For a user manual for a non-technical client, you'll need a little more "padding", but the topic flow is similar:

Cover page and TOC

The usual cover page, with "User Manual" as prominent title. Include a brief Table of Contents. This document isn't that long, so just entries for the key things that client wants to do (roughly matching the sections below) will suffice. If you have room, you can just put this neatly at the bottom on the cover page for easy reference. Otherwise put it right under the cover page…and don't skip a page after, just start the intro right under it.

Introduction

This is simple, just one paragraph. You don't need the broad motivations of previous documents, just dive right in, e.g., "We are pleased that you have chosen <sys_name> for your business needs. <Sys-name> is a powerful system for <overall statement of what it does> that has been custom-designed to meet your needs. Some of the key highlights include: <bullets of 3-5 coolest features>. The purpose of this user manual is to help you, the client, successfully install, administer, and maintain the <sys-name> product in your actual business context going forward. Our aim is to make sure that you are able to profit from our product for many years to come!" Or something along those lines.

Installation

Here you are going to walk through, step-by-step, what it would take to install the product on an appropriate platform. A nice segue to start it off might be something like "As part of final delivery, the <sys-name> system should have been installed on a platform of your choice. Over time, however, you may want to move to a new platform or re-install the product." Then you transition to a careful description of how to install it. Write it for a halfway technical user. If your client is completely computer illiterate, you can assume that he/she will get someone that knows the basics to help. So, for example, if you need to open a terminal window on the machine, you don't need to explain how to do that. But you would have to give exact commands, etc, needed to install the product. A f*ew screen snapshots can be really useful* here, to show what the client should be seeing/doing.

Configuration and Daily Operation

In the last section, you hopefully ended with the client being able to log into (or connect to, or whatever) the installed product. In this section, you will detail whatever tasks need to be done in order to get the product deployed and operational. Details depend on individual product, but might include steps like "Configure admin user profile and password", "Create user accounts", etc. This is the bulk of your user manual, and should simply cover any tasks the client may need to do (while consulting this manual) on a regular basis to operate the product.

Maintenance

This is a separate section to highlight specific activities that the client should do in order to maintain the long-term health of the product. This could include how to do/schedule backups, how to clean out the logs every once in awhile, etc. Imagine what "dirt" will accumulate in your product over time…and give instructions on how to deal with that. Be sure to indicate your "maintenance schedule" clearly, i.e., how often the client should do these things, how big a logfile can get before it needs truncating, whatever. Some products will have a longer section here, other more "low maintenance" products might have just a few reassuring lines. **One specific thing you'll want to include is access to your source code.** That usually means (a) making the client the owner of the repo that you used or otherwise transferring the repo to them, with clear documentation of this in the manual; and (b) providing the client with a USB stick with your repo copied onto it.

Trouble-shooting

We have all, at one time or another, been saved by a well done “trouble-shooting” section in a manual, so the purpose here should be clear. The idea is that the manual body says a lot about how to operate the system, but not much about what might go wrong. Thoughtfulness here is key; a trouble-shooting section that covers problems that never actually happen is only frustrating. Think about what has gone wrong you during testing, what could be broken by the client or by a change in the environment. Think about the *dependencies* within your product, and between your product and the environment. Then give fixes for each. In real life, this section would be refined over time after product deployment, based on tech support feedback. Lacking that, you’re careful thoughtfulness will have to do here.

Conclusion

This just formally closes the document. Wish the client many happy years of productive use of the product, express your joy at having been able to help. Maybe end the section with “With best wishes from your <sys-name> developers:” and then list out your names. If you want to be nice, include your emails and a statement that “While we are all moving on to professional careers, we would be happy to answer short questions in the coming months to help you get the product deployed and operating optimally in your organization”.

The Product Delivery Check-off sheet

In the excitement to deliver a product and move on to the next project, it’s easy to forget some of the steps above…both for you and for the client. Only later, when the excitement is over and your team is long gone, does the client discover that they are unable to install the product, that they are missing key privileges, or that the user manual is missing. The Product Delivery Check-off sheet is designed to avoid these problems. It protects the client from ending up with a product that they can’t deploy (what a waste after a whole year of effort!); and it protects the team by documenting their due diligence in delivering the product and (critically) the client’s sign-off for having approved and received the product.

The Product Delivery Check-Off worksheet is posted to the Capstone website. You should bring a hardcopy of this with you when you meet with the client for the product hand-off, and use it as an agenda for that meeting, simply walking through all the tasks listed and getting the client’s comments and check-off on each.

Assignment specifics: What to do

The outline above should have given you a good idea of what needs to happen for product delivery, but here is a list of specific things you’ll need to do:

1. Create your User Manual. Review it carefully to make sure it is complete and understandable. Make your Ceramics Major roommate read it and see if it seems sensible to him. Print a final version, and bind it in a nice professional sleeve. Tape your thumb drive with your codebase, plus electronic versions of all team documents (including the user manual!) on the inside cover.
2. Send your client the User Manual electronically, at the time that you arrange the delivery meeting, and ask them to look through it before you meet.

3. Arrange a formal meeting with your client, preferably in-person, to do the delivery. Teams with clients outside of Flagstaff can arrange a formal Skype meeting.
4. Conduct the meeting, walking through the check-sheet and explaining every item on it to the client’s satisfaction. **The client must sign off on your check-off sheet!**
5. Get a hardcopy of your User Manual and the signed check-off sheet to your mentor.
6. Go out as a team and get a beverage of your choice to celebrate a job well done!

Deliverables

- Hardcopy of your User Manual, **with the signed original sign-off sheet attached**. Delivered to your team mentor for grading by the end of finals week.