

Final Project Process Checklist

Lawyering in the Age of Smart Machines

Fall 2016 - Suffolk University Law School, David Colarusso & William Palin

This is a companion to the [Final Project Grading Rubric](#). It's intended to help jump start your work by providing you with concrete steps tied to process categories in the rubric. It is not a prescriptive document in that you do NOT have to follow these steps. Your project will be evaluated based on the criteria laid out in the rubric, NOT how well you conform to the steps below. That being said, if you're having a hard time getting off the ground, these steps are as good a place to start as any.

Framing. *How well does the student define a problem facing legal practitioners or consumers, including a definition of relevant stakeholders / users?*

What is the problem; who is having it?

- ☐ Identify a problem facing legal practitioners or consumers of legal services.
- ☐ Can you see the outlines of a solution involving a tool or tools you know how to use? You don't have to have a fully-formed vision, just the sense that it's possible. **If not, rethink your problem.** We've primarily dealt with two types of relevant tools in class: (1) expert systems--interactive decision trees; and (2) document automation--automated *find-and-replace*. Of course, these can be used in conjunction with each other and other tools. You might want to consider the following: QnA Markup, A2J Author, Neota Logic, and HotDocs. Everyone should have enough familiarity with QnA to use it as an expert system in aid of document automation, but don't feel confined to this tool if you are familiar with others.
- ☐ **Are you confident in your ability to build a working solution between now and December 20th (the last day of the exam period)? If not, rethink your problem or your tool(s).** Note: both professors Colarusso and Palin are available to help you troubleshoot during your development process. So, e.g., if you're not sure how to make QnA do X, they can help.
- ☐ Document your problem and proposed solution in your Project Biography. A few short lines will do. Identify the people who face The Problem as well as the people who would use your solution as these are not always the same people.

Research. *How extensive is the student's research of existing solutions?*

What could fix it?

- ☐ Research if anyone has solved or is working to solve this problem. Can you learn anything from these attempts? What are they doing that looks promising? Problematic? Why is your solution worthwhile? Is it less expensive? Is it easier to use? Is it more appropriately tailored to The Problem?
- ☐ Find someone who has The Problem or works with people who have The Problem. Talk to them about what they would like to see in a solution.
- ☐ Document your research and discussion(s) in your Project Biography.

Expert System Creation ProTips

1. Remember, you are breathing life into a flowchart. Everything follows from that.
2. Identify and list your desired end states. For example, if your system is trying to answer a question with a Yes or No answer, Yes and No are your two end states. If your system is trying to direct people to a single resource out of many, the end states are each of the resources to which you could direct a user. If your system is populating a set of documents, your end states are the specific document or documents you could create for a single user.
3. Determine what questions you need answered in order to determine a user's end state (e.g., which documents they will get). When creating documents, these questions will include those needed to fill in all of the form's "blanks." See Document Automation ProTips below.
4. Create a script in which your users can get to the end states in as few questions as possible. Placing your end states and questions on Post It Notes or index cards can be a good way to work through this. For example, place Post Its for your end state on one end of a whiteboard. Then arrange your questions to one side, drawing a path through questions that makes the most sense. This will require a lot of play.
5. Once you have a flow chart, translate its logic into your expert system.

Document Automation ProTips

1. Create a version of your final document.
2. Identify those places in the document where text will differ. Consider if these bits of text will be in every potential version of a document or only some. For example, if you are automating a contract, the party names are likely to be in every document, but specific clauses may be contingent on a set of facts.
3. Create a list of those bits of text, like names, that will appear in every document. Consider these to be your variables (i.e., the subject of a future find-and-replace).
4. Consider if contingent text, like clauses, are better addressed by treating them as variables OR if you would be better off having multiple versions of the document with different contingent text. If the former, add them to your list of variables from #3 above.
5. If working with QnA & Word
 - a. Rewrite your list of variables so that each variable is a single "word," comprised only of letters, numbers, dashes, and underscores.
 - b. Create an Excel sheet with a column for each of your variables and at least one row of dummy values. Save your work.
 - c. Open or create a final version of your document (you will have to do this for multiple documents if you have the need for more than one template). Use the Excel sheet as a mail merge data source, and place your variables in the document as needed. Save your work, preferably after returning the document to not prompt for mail merge on open (Colarusso can show you how). See steps 2-3
 - d. You can now use this Word file as your template file. See http://www.qnamarkup.org/syntax/#docx_docs and https://github.com/colarusso/docx_webmerge. Note: it will need to be uploaded to our class website in order to work with docx_mailmerge. To do this just put it in a project file under /students on our website. Here's the [how to](#).
6. Otherwise, take this information to create your document template with the tool of your choice.*

*For those who want HotDocs, **It works for 120-day trial version of HotDocs Developer, and it is available at <http://www.hotdocs.com/educational-download>.**

The password is **k3gh87ht**.

You can choose either Professor in the signup process

Ideation, Prototyping, and User Testing. *How well does the student explore the available space of potential solutions? How rigorously does the student engage in user testing, and how realistic are such tests?*

Can my fix actually deliver?

- ❑ Think of several ways to address The Problem. These may include using different tools or taking different approaches with the same tool, and they may involve minor or major differences.
- ❑ Build prototypes for some subset of these options and test them. Note: your initial prototypes can be as simple as a [storyboard](#), and testing can be as simple as getting a potential user's feedback.
- ❑ Document your work in your Project Biography.

Refinement. *How well does the student integrate feedback from user testing into subsequent versions of their solution?*

Can I make it better?

- ❑ The point here is that you should be approaching your work as an iterative process: design, test, repeat. You might do this without talking to anyone. That's fine, just document what you do.
- ❑ As noted above, your initial prototypes don't have to involve working code. However, as you loop through the design process (the *repeat* in *design, test, repeat*), you should start to focus on a single prototype and improve upon it until it becomes your final product. In the end, [there can be only one](#).
- ❑ Document your research and discussion in your Project Biography.

