Wednesday, 2/1 - Business casual

* What are the “existing high-level requirements” that were mentioned?
* "In a modern language" - clarification of this, what language (any languages we should avoid?)
* If writing an automated system, do they know of any tools for that
* "goal is based on these two tools" -
* What does it mean by "our tools"
* Are you looking for open source/free software tools? **Must be free**
  + ~~If not, what is your budget?~~
* What are you looking for as a final deliverable? (What is the outcome that you are looking for?)
  + GUI / command line tool to enable conversion of any project?
* Will we be able to look at/work with the tools being used in order to better work with them specifically?
* Importing libraries can cause security or licensing issues
* How well formed will the input be?
* Are there any coding standards/conventions followed by the company that we should also follow for our project?
* Extension to the previous question: Are there any standards as to how documentation should be completed for the company outside of what we are required to do in class?

Notes from Sponsor meeting:

* Manual for CMS 2, very old, will be sent out
* preferably **Python** (but any are acceptable)
  + Would love to be able to export to excel/csv file
* Idea: rule-based framework that we can build on as we go along (rule based on grammars?)
* Final Deliverables:
  + Prototype product that we created
  + Research other products that could be used
  + Human Interactive
  + Research Red Hat Linux

Notes from Sponsor Meeting:

Craig Sobieralski (craig.sobieralski@asrcfederal.com) and Chris Bartley (christopher.bartley@asrcfederal.com)

Meetings would be on campus (in class?)

If meeting becomes important, can meet at Rowan.

*Manual for CMS2, need to read*

* Conversion, find new tools, automate?
* Allowed to use libraries that exist
* Output has to be **exactly** the same as original output
* MS can't use existing converter because tend to break, don't work
* Code is going to be weird (because FORTRAN sucks)
* They have the original algorithm
  + Planning on giving us the input they were going to use and output looking for
* There's no budget - any third party tools would have to be free
  + Free versions, demo could be used to suggest expenditure
* Looking more for Python, caveat any of them are acceptable (if Java is that much easier, can use that)
* Successful product - hoping for both
  + Looking at coding ad present a prototype, but also look at available software, present both
  + Definition of done should include multiple options to complete the task
  + Include the time/difficulty and potential cost
* Final deliverable - presentation on what code base we're using
  + No demo unless we have something to present
  + Lessons learned, what we tried, other software we found
* Can code it all on Windows, just have to make sure we can bring it over to Linux/Unix
* Looking for an automated tool, as we are not going to be given the actual code of the tools
* There are multiple tools - starting out with first tool, finishing the others may depend on the speed in which we finish
  + Should be a more general product, as we might need it to work for multiple tools with different output
* Human interactable vs. background - most likely human interacting, users don't want something completely different
* Partial solution/framework solution is fine, but should aim for something more complete
* Auditing tools, making sure developers are kept honest, reports are sent out
* Function over style/appearance
* Progress reports out of Trello? Telling sponsors where we're at
  + What was accomplished, general high-level lessons learned at the end of sprints, obstacles
  + What we run into, questions concerning what we wrote or have so far
* Sprucing up the output would be good as an add-on, even if not fully functioning
  + Alternative output options?
* Coming up with a GUI would work well, currently command-line based
* Advice, going into it
  + Enjoy this, only time we get to play around
  + Think outside the box
* Agile model of development

Daily Scrum (More of a Sprint Planning Session) - 2/2/17 - 3:30pm in CS Lab

* Going to seek to solve problem using Python
  + Rule based conversion program
    - From FORTRAN to Python
  + We will try to create the rule-based conversion program and if that doesn’t work out, then we will manually convert from Python to FORTRAN.

Sprint 0 Review Notes

* Definition of Done: be specific
  + scalable/extendable
  + UML / sequence diagrammed
* Idea for design:
  + Abstract parser class
    - Should be extended for each language
  + Intermediate output is parse tree (new and old version of the files) for the given language (especially important to split into statements)
  + Given parse tree, run diff style function on the new and old version and generate reports that include number of insertions, deletions, and modifications for code, comments, etc.
* Find git APIs for Python?
* Check out how Unix diff command works
* Input is set of CMS-2 files, output is report