**Meeting Minutes**

**Date:** January 28, 2014

**Start Time:** 7:15pm

**End Time:** 8:05pm

**Members Present:** Drew Aaron, Michael Beaver, Clay Boren, Chad Farley,

Andrew Hamilton, Travis Hunt, and Dr. Patricia Roden

**Members Absent:** N/A

**Topics** **Discussed**

* Client questions #2
* Screen captures and prototype mockup
* Post-client-meeting sessions
* Logging work hours

**Decisions and Actions Taken**

The team met with the client (Dr. Roden) to discuss and answer a second round of questions. The questions and the client’s responses are attached.

The client reviewed a few screen captures and a rough mockup of the new IDE. The client is dissatisfied with the memory format. She wants the memory to be formatted like it is formatted in ASSIST/I. The registers also need to be properly numbered, and the Program Status Word (including Condition Code) should be made available. The client also wants to see different color schemes (preferably no blue).

The client has decided that the team is *not* ready to propose a programming language at this time. The client approved the team’s choice to follow a Hybrid RPM-Waterfall method process lifecycle.

The team needs to ask the client about the syntax highlighting (color schemes, syntax to be highlighted, etc.) at a later date.

The team has instituted a new post-client-questions meeting policy. After each client-questions session, the team will meet to discuss the client’s responses to the questions. These meetings will allow team members to discuss any issues raised by the client’s responses and allow the team members to improve their understanding of the project requirements.

The team has instituted a policy of logging *all* hours worked while contributing to the project. This will allow the team to keep an accurate record of contributions by team member and by time invested.

Travis has created a OneNote common notepad for the team to take common notes that can be saved and updated via SkyDrive. Adoption of this OneNote notepad is still pending.

It has become apparent that the team needs to begin backend work as soon as possible, probably as soon as the programming language is approved. The parser portion of the backend has grown more complex after some analysis of the original ASSIST/I.

The team will begin working as a team, as soon as possible, to design a preliminary prototype. This is necessitated by the chosen Hybrid RPM-Waterfall Method lifecycle process.

The next team meetings will be Friday, January 31, 2014 at 4:30pm at the Christian Student Center and Monday, February 3, 2014 at 6:00pm in the Collier Library basement.

**Supplementary Information**

**Questions answered by the client:**

1. When will the subset of ASSIST/I instructions be made available?

**Drew Aaron:** Answered.

**Michael Beaver:** On ANGEL.

**Clay Boren:** On ANGEL.

**Chad Farley:** On ANGEL as of January 28, 2014.

**Andrew Hamilton:** On ANGEL.

**Travis Hunt:** Answered on ANGEL.

2. What is the “identifier” for the “Final Run” command in ASSIST/I?

**DA:** Your name for the output file.

**MB:** The programmer’s name. Be sure to implement.

**CB:** [no response]

**CF:** Programmer’s name or “handle.”

**AH:** Puts name at top. Be sure to implement.

**TH:** Implement it. It is used for the .PRT file.

3. What is the maximum size of memory for the IBM/360 with ASSIST/I?

**DA:** Option for total size in bytes.

**MB:** Memory is to be displayed as Address—Memory Contents—Character Representation. The maximum displayable memory is tied to the maximum memory size option. There is an error message when the maximum memory is exceeded. This error should be displayed to

the user. This can be tested using the multiple bank account program from CS 310.

**CB:** 2700 bytes.

**CF:** Memory is displayed as Address—Memory Contents—Character Representation. Memory size can be toggled with options.

**AH:** Maximum size (in bytes) option.

**TH:** The option controls size of display; an error is given if memory size is exceeded (see multiple bank accounts program).

4. Does the GUI need an input field, such as the one in MARIE?

**DA:** Test it in ASSIST/I. Do it like that.

**MB:** Users *must* be able to provide input. There is a special $ENTRY command (see the User Manual for ASSIST/I). Be sure to test ASSIST/I with console input.

**CB:** Test out like ASSIST/I.

**CF:** Yes. Handle like the ASSIST/I would. Run tests to determine.

**AH:** Yes. Test on ASSIST/I.

**TH:** Test console input on ASSIST/I first.

5. Do you want output to be changeable (ASCII, Hex, Decimal) as with MARIE or as ASSIST/I would normally represent it?

**DA:** No.

**MB:** No. Format output with straight hexadecimal, as with ASSIST/I.

**CB:** Just hexadecimal.

**CF:** [no response]

**AH:** No.

**TH:** Just like ASSIST/I.

6. Would you like a “view .PRT file” button or option?

**DA:** Only manual open.

**MB:** Only manual open button.

**CB:** Use manual open.

**CF:** Manually open the .PRT only.

**AH:** Only manual open button.

**TH:** Only manual button. Do *not* have an auto-open option.

7. Where would you like the output to the screen to be? Would you like it to be the traditional command prompt style or like a modified version of MARIE?

**DA:** Like a modified MARIE—just print output.

**MB:** Output from output command (i.e., XPRNT) in a way like modified MARIE.

**CB:** Use modified version of MARIE.

**CF:** Output using the modified version of MARIE with a button to open the .PRT file.

**AH:** Just the output like MARIE.

**TH:** Automatically show XPRNT. There will be a button for the detailed .PRT file.

8. How do you want memory dumps from crashes to be displayed?

**DA:** Like ASSIST/I. Be able to handle XDUMP. See code below.

**MB:** Memory dump is formatted just like if it is run normally (see #3). Fatal crash errors are saved to the .PRT file, like in ASSIST/I. There *must* be the capability to use the XDUMP command. The XDUMP command dumps registers or memory (depending on the arguments) and keeps running the program. See the User Manual.

**CB:** Use XDUMP.

**CF:** Traditional memory dumps from ASSIST/I. Be able to use the XDUMP command.

**AH:** Save in MARIE. Include the XDUMP command.

**TH:** Just to the .PRT file. Include the XDUMP command.

9. On the detailed project description, which specific features are required? Which specific features are optional?

**DA:** Other useful features: online manual only required. Disassembly is optional.

**MB:** Online help is *required*. Disassembly is *not* required. Proper tabbing, code completion, and hotkeys are *not* required.

**CB:** Online help and disassembly are optional.

**CF:** [no response]

**AH:** Online help *required*. Ask about syntax highlighting.

**TH:** Online help *required*. Ask for guidelines on syntax color.

10. What exactly do you mean by “set tabs to implement correct code layout” in the detailed project description?

**DA:** Tab will go to next column (optional).

**MB:** *Not* required.

**CB:** [no response]

**CF:** Optional: Tab should push cursor to next programming column.

**AH:** Nice but not necessary.

**TH:** Tab by column stops.

11. What are your thoughts on the following names: ASSIST/J; COBRA; ASSIST/II; ASSIST/I++; and, ASSIST/101?

**DA:** ASSIST/UNA or UNAssist.

**MB:** ASSIST/UNA or UNAssist. Incorporate “UNA” into the name somehow.

**CB:** ASSIST/UNA or UNAssist.

**CF:** ASSIST/UNA or some variant.

**AH:** ASSIT/UNA or UNAssist.

**TH:** ASSIST/UNA or UNAssist.

**General notes from Client:**

* Program Status Word (PSW) holds the Condition Code (CC)
* The client does not like a lot of blue.
* The client does not like the MARIE memory representation.
* The client accepted the team’s proposal to follow the Hybrid RPM-Waterfall Method.
* Memory is to be represented like in ASSIST/I.
* What is displayed in memory is tied to the maximum memory size in Options.
* If memory is exceeded, stop and give an error message telling the user what to do.
* The client wants memory to be represent in the GUI, not in tabs.
* The data in the registers must be represented in hexadecimal.
* The memory formatting will need to be changed to reflect ASSIST/I’s formatting.
* ASSIST/I uses EBCDIC, not ASCII!

**Code from Drew Aaron for #8 from the Client Responses:**

NAME CSECT

L 1,16(,15)

L 2,20(15)

AR 1,2

ST 1,50(15)

BCR B’1111’,12

DC F’4’

DC F’4’

DS F

END NAME