

Computer Science

MONTANA TECH OF THE UNIVERSITY OF MONTANA

MINUTES OF THE COMPUTER SCIENCE ADVISORY BOARD MEETING 23-24 March 2000

Opening and welcoming comments were made by Keith Olson. Those present included Russ Lewis, Celia Schahczenski, Keith Olson and Donna Kay Huntoon from Montana Tech, John Thurmon from Montana Power in Butte, Davis Almanza from Computers Unlimited in Billings, and Bruce Lowther from Micron. Bruce came in place of Michael Ziegler. Michael couldn't make it so he asked if Bruce could come instead.

There was a review of the last meetings minutes. A set of by-laws was created. The three by-laws that were agreed upon were:

- That the Advisory Board should meet once a year.
- The Board should be able to make recommendations that will be acted upon.
- There should be continuity. They felt there should be a three-year term limit.

The current curricula were discussed. It was decided to start with a clean slate. The general education requirements were discussed. There is going to be a change in curriculum and there were questions about what was required and what wasn't.

There was some lengthy discussion about the different languages taught. Keith and Russ teach some language other than Java and C or C++. Russ tries to add a little bit of Perl to his class. He would like to teach Advanced Java. Some CDs have different languages on them. Bruce suggested that two languages are good. He suggested that there be a typed and not typed language so the students can see the difference.

The UNIX, C and C++ class was discussed. It was suggested that these three languages were just too much for a 3 credit class. The suggestion was made to make UNIX and C one class and C++ as another class.

There was quite a bit of discussion about cutting out a math course and putting in another computer science course. The idea came up to cut Elementary Differential Equations. The suggestion was made to direct students to Independent studies to keep them from the "garbage" classes....which were described as classes the students take to make up the 30 hour credit requirement for a double major.

Keith spoke of the Software Engineering class he teaches. He said that he has the kids keep a record. This allows them to keep track of the amount of time the project took and it helps to point out the same errors. This enables the student to make predictions of how much time it will take them to complete a project. This brought about the discussion that there are 20 or less people who have a Bachelor of Software Engineering degree. None of these programs have been accredited yet.

The proposal was made to put in a Bachelor of Science degree in Information Technology (IT). Keith fears this may result in a "watered down" degree. Bruce said that out of 500 people he interviewed with IT degrees, only one was hireable. He said she was hireable only because she had worked at learning more than just what was taught. He feels this is a very

weak program. Most people who want an IT degree want to go into e-commerce, web page design and other such careers. The question was asked if the IT program would be similar to other C.S. programs. WSU dropped Calculus from its IT program. But they still have discrete math. John Thurmon said the IT program puts the computer science degrees in jeopardy. He said it is his opinion that IT people are not technical just glorified tech writers. They don't have any language background. Bruce felt that it should be an Associate degree instead of a Bachelor degree.

Bruce made a proposal for Software Engineering. The proposal included but is not limited to:

- SE gives the opportunity to expose the larger package that is available out there.
- Use G-make and expand SE in this way.
- Use E-max and expand SE in this way also.
- Since SE requires testing and more concentration, there should be more emphasis on analysis, design, etc.

On Friday, 24 March 2000, Keith opened the meeting by telling us that we have just received a \$1000 scholarship from Texaco. Those present on Friday morning were, Keith, Celia, Donna Kay, John, Davis, and Bruce. Russ Lewis joined us for lunch and was available for a bit afterward and then he had to go to class.

We started the meeting by discussing Skills-Outcome Desired, Computer Science vs. Software Engineering. What do the students want??? Knowledge. What do the students need to know when they graduate??? How to interview.

Under the Knowledge category of what students want, the following were discussed:

COMPUTER SCIENCE DEGREE

- Language Skills—What languages do they need to know?
 1. Assembly
 2. C++
 3. SQL
 4. C
 5. Perl (untyped and not strongly typed; functional)
 6. UML (How do we work this in?)
 7. Visual Basic??
 8. ABAPP (Advanced Business All-Purpose Programming)
 9. Java (not mature enough)
 - a. Object Oriented Programming (don't put language in title; Visual Basic)
 - b. Procedural Programming (Engineers want C or FORTRAN)
- Data Structures Course (strong emphasis on arrays, linked lists, etc.)
- Algorithms
- Networks
 1. How to do it (OSI, newer ones)
 2. Theory Protocols
 3. Security (Do students need to know about security? John, Davis and Bruce all said YES!)
- Architecture (heap, stack, pass-by value, pass-by reference, related to Object Orientation. Taught mostly in C.S. 215 and C.S. 440)

- Operating Systems (semaphores, threads IPC, signals, windows message loop)
- Sockets
- Real Time (should this be taught in Software Engineering??? Again YES by our guests)
- Optimization (get program to work before optimizing...a high priority, Maintainability....do a code walk through...this is something for SE)
- Exception handling
- Compiler Construction
- Object Oriented concepts (Polymorphism, inheritance, encapsulation....SE)
- Constructors, Destructors, etc. cf, pbr, pbv taught in C. S. 331 and 332)
- Data Base Normalization (taught in C.S. 365)

SOFTWARE ENGINEERING DEGREE

- Processes
 1. Source code control (Version control. Have they built and used a library?)
 2. Teams (How do the students work in a team environment? Suggestion was made about setting up an internship so the students could see what it is like to be in a "real" working environment. Maybe have an internship with a company where they have to do a real project for the company)
- Code Reuse/Modification
- Conversion to new systems without interruption
- Feature negotiation (important for a student to know in an interview)
- Schedule planning
- UML - CMM, PSP, TSP, structured charts (What kind of tools? Charts? YES)
- Maintainability
- Upward compatibility (Planning for change)
- Multiple teams have to interface with several other team projects, unit testing, interval testing.

Networking was discussed. Troubleshooting and a discussion of how many servers are needed ensued. It was felt that networking has to more with telecommunications. System design (not sw) was discussed as well as graphics. All three of our guests said that graphics is fun but not used in their companies. But that it is important to those who do use it in their companies.

There was quite a bit of discussion regarding the Software Engineering degree.

- Formal method was discussed as a math class requirement for this degree.
- There is a problem with credit hours
- How important are the specialized electives?
- Validation and Verification

The General Education requirements received quite a bit of discussion. The results of that discussion are:

- Communications are important. Students should have to present an oral presentation and this experience should be used in all other classes.
- Economics is very important especially cost/benefits and supply/demand
- Ethics should be a one credit required class

- Continuing education such as reading the proper journals, attending seminars, etc

There was a lot of discussion about which degree some of these things should be under. Windows messaging, exception handling and threads should be taught in the Operating Systems part of the courses. Sockets, real time, optimization, object-oriented concepts and patterns should be taught in Advanced Programming. Modification of code, system conversion, feature negotiation, UML should be taught in Software Engineering. System Design and Validation and Verification should be taught in the Computer Science degree program.

Some of the conclusions to these discussions were:

- Economics should be a general education requirement
- Ethics and a Senior Seminar should be combined
- Communications in all courses with a strong emphasis on speech and writing skills
- Additional programming language in Data Base
- Replace Elementary Differential Equations with Methods of Proof
- Eliminate Artificial Intelligence and Numerical Computing
- Do away with the professional electives
- Redesign Options (be careful of the 30 credit outside requirements)
- Tech's credibility may suffer if we offer an IT program (our guests felt it may weaken the C.S. program because the program would be diluted)
- Networking should be taught in the Fall of the Junior year
- Compilers should be taught in Fall of Senior year
- Senior Seminar should be in the spring of last semester
- Move the Database class back from the Junior year to the Sophomore year
- Move Humanities from the Sophomore year to Spring of the Senior year

The decision was made to change the curricula to show the following changes:

Freshman Year

FALL	SPRING
C.S. 210	C.S. 211
Math 152	Math 153
Engl 104	HSS 121
SS	SS
	Science

Sophomore Year

C.S. 331	C.S. 332
Math 325	C.S. 365
C.S. 316	Math 222/223
Math 251	Math 331
	Elective

Junior Year

C.S. 325	Hum
C.S. 451 (Networking)	C.S. 326
Science	C.S. 340
Prof. Elective	Engl. 391

Prof. Elective

Senior Year

C.S. 438
C.S. 215
Compilers*
Project*
Prof. Elective

C.S. 401
Senior Seminar*
C.S. 440
Project*
Prof. Elective

*These are classes that we would like to implement but have not been and at this time do not have a number assigned to them.

The meeting concluded with everyone agreeing that March is a good month to have the meeting in. It was also agreed upon that there should be an agenda sent out to those we invite and it should start in January of 2001.

Respectfully submitted,

Donna Kay ☺
Secretary
Math/C.S. Department