B.Sw.Eng. @ UWaterloo

An undergraduate degree program in Software Engineering at the University of Waterloo, jointly sponsored by the depts of CS and ECE.

Ad Hoc Committee Members

- Rudi Seviora [ECE]
- Jo Atlee [CS]
- Bruno Preiss (him)
- Ric Holt
- Paul Dasiewicz
- Anne Pidduck
- Kostas Kontogiannis
- Mike Godfrey (me)
- Stefan Leue
- Grant Weddell
- Ajit Singh
- Farhad Mavaddat

Background and Motivation

- Software runs the world, for better or worse.
- Software systems comprise major assets of large corporations.
- Software is become even more pervasive!
- Industry is crying out for more, better help.

Why not ...

- B.Eng./B.A.Sc. in ECE
  - not nearly enough “software content”
- B.Sc./B.Math. in CS
  - no “engineering” mindset
  - not enough coverage of some engineering topics

Why not ...

- B.Eng. + B.Sc.
  - Not enough coverage of “software engineering”
  - HCI? Project management?

Program Design Goals

- A true software engineering curriculum!
- Graduate emerge prepared for life as a software professional.
- Curriculum covers the scientific and mathematical foundations.
- Curriculum includes engineering science and engineering design.
Program Design Goals

- Curriculum exposes students to ethical and societal issues of SE.
- Curriculum permits electives (technical and general).
- Graduates should be able to apply their knowledge to specific problems and produce solutions.

Curriculum Design

- Proposed curriculum is based on many sources:
  - ACM
  - IEEE
  - SEI
  - CSAC
  - CEAB
  plus interactions with CS and ECE curriculum committees.

Curriculum Outline

- Core
  - mathematics
  - natural sciences
  - digital systems
  - computer science & engineering
  - software engineering
  - complementary studies

- Electives
  - general
  - linkage
  - advanced
  - technical

Core Mathematics

<table>
<thead>
<tr>
<th>Course</th>
<th>Mathematics</th>
<th>Linear Algebra</th>
<th>Discrete Mathematics</th>
</tr>
</thead>
<tbody>
<tr>
<td>1A</td>
<td>Calculus 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1B</td>
<td>Calculus 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2A</td>
<td>Probability &amp; statistics</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2B</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3A</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3B</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4A</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4B</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Core Natural Sciences

<table>
<thead>
<tr>
<th>Course</th>
<th>Natural Sciences</th>
</tr>
</thead>
<tbody>
<tr>
<td>1A</td>
<td>Mechanics</td>
</tr>
<tr>
<td>1B</td>
<td>Electricity &amp; magnetism</td>
</tr>
<tr>
<td>2A</td>
<td></td>
</tr>
<tr>
<td>2B</td>
<td></td>
</tr>
<tr>
<td>3A</td>
<td></td>
</tr>
<tr>
<td>3B</td>
<td></td>
</tr>
<tr>
<td>4A</td>
<td></td>
</tr>
<tr>
<td>4B</td>
<td></td>
</tr>
</tbody>
</table>

Core Digital Systems

<table>
<thead>
<tr>
<th>Course</th>
<th>Digital Systems</th>
</tr>
</thead>
<tbody>
<tr>
<td>1A</td>
<td>Digital circuits &amp; systems</td>
</tr>
<tr>
<td>1B</td>
<td>Digital computers</td>
</tr>
<tr>
<td>2A</td>
<td></td>
</tr>
<tr>
<td>2B</td>
<td></td>
</tr>
<tr>
<td>3A</td>
<td></td>
</tr>
<tr>
<td>3B</td>
<td></td>
</tr>
<tr>
<td>4A</td>
<td></td>
</tr>
<tr>
<td>4B</td>
<td></td>
</tr>
</tbody>
</table>
### Core Computer Sci. & Eng.

<table>
<thead>
<tr>
<th>Core Courses</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1A</strong> Developing programming principles</td>
</tr>
<tr>
<td><strong>1B</strong> Principles of computer systems</td>
</tr>
<tr>
<td><strong>2A</strong> Foundations of sequential programs</td>
</tr>
<tr>
<td><strong>2B</strong> Algorithms &amp; data structures</td>
</tr>
<tr>
<td><strong>3A</strong> Control structures</td>
</tr>
<tr>
<td><strong>3B</strong> Real-time operating systems</td>
</tr>
<tr>
<td><strong>4A</strong> Computer networks &amp; database systems</td>
</tr>
<tr>
<td><strong>4B</strong> System performance evaluation</td>
</tr>
</tbody>
</table>

### Core Software Engineering

<table>
<thead>
<tr>
<th>Core Courses</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1A</strong></td>
</tr>
<tr>
<td><strong>1B</strong></td>
</tr>
<tr>
<td><strong>2A</strong></td>
</tr>
<tr>
<td><strong>2B</strong> Software abstraction &amp; specification</td>
</tr>
<tr>
<td><strong>3A</strong> Software project management</td>
</tr>
<tr>
<td><strong>3B</strong> Software requirement analysis</td>
</tr>
<tr>
<td><strong>4A</strong> Software design &amp; architectures</td>
</tr>
<tr>
<td><strong>4B</strong> Software testing &amp; quality assurance</td>
</tr>
</tbody>
</table>

### General Electives

- 3 courses
  - Ideally unconstrained to allow student to customize the degree to an application area of interest to the student
  - Initially constrained to satisfy CEAB natural sciences criteria

### Linkage Electives

- 5 courses
  - Communication
  - Societal issues
  - Business issues
  - Reasoning methodologies
  - Humanities & social sciences
  - plus Engineering economics (core)

### Advanced Electives

- 2 courses
  - Fourth-year CS or E&CE courses
  - Chosen to expose students to one or more significant application areas

### Advanced Electives

- Compiler construction
- Information system management
- Introduction to AI or Applied AI
- Introduction to computer graphics
- Distributed and network-centric computing
- Embedded software systems
- Numerical Linear Algebra
- Numeric Computation for Dynamic Simulation
Technical Electives

- 3 courses
  - Selected third- and fourth-year technical courses, or
  - Advanced electives
- Offered by CS or E&CE
  - No overlap with SE program courses

Program Skeleton

<table>
<thead>
<tr>
<th>math</th>
<th>sci</th>
<th>GE</th>
<th>LE</th>
<th>dig</th>
<th>comp</th>
<th>softw</th>
<th>TE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1A</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1B</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2A</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2B</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3A</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3B</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4A</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4B</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Program Sequence - Year 1

- Calculus 1
- Linear algebra
- Mechanics
- Discrete mathematics
- Developing programming principles
- Calculus 2
- Electricity & magnetism
- Logic & computation
- Principles of computer science
- Digital circuits & systems

Program Sequence - Year 2

- Probability & statistics
- Foundations of Sequential programs
- Digital computers
- Managerial & engineering economics
- General elective
- Algorithms & data structures
- Software abstraction & specification
- Dynamic systems analysis
- Linkage elective
- General elective

Program Sequence - Year 3

- Software project management
- Control structures
- Human-computer interaction
- Linkage elective
- General elective
- Software requirement analysis & spec
- Real-time operating systems
- System performance evaluation
- Technical elective
- Linkage elective

Program Sequence - Year 4

- Software design & architectures
- Computer networks & security
- Database systems
- Technical elective
- Linkage elective
- Software testing & quality assurance
- Technical elective
- Technical elective
- Linkage elective
- Technical elective
CEAB Accreditation Units

<table>
<thead>
<tr>
<th>category</th>
<th>actual</th>
<th>required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mathematics</td>
<td>298</td>
<td>195</td>
</tr>
<tr>
<td>Basic science</td>
<td>212</td>
<td>225</td>
</tr>
<tr>
<td>Engineering science</td>
<td>339</td>
<td>225</td>
</tr>
<tr>
<td>Engineering design</td>
<td>565</td>
<td>225</td>
</tr>
<tr>
<td>Eng. sci. &amp; design</td>
<td>934</td>
<td>900</td>
</tr>
<tr>
<td>Complementary studies</td>
<td>281</td>
<td>225</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td><strong>1815</strong></td>
<td><strong>1800</strong></td>
</tr>
</tbody>
</table>

Social Issues of Curriculum Design

- Initially, some resistance from within CS and ECE, but good will has won the day
- Some fine tuning still needed.
- Issues:
  - How much of topic XXX vs. YYY?
  - New courses or re-used courses?
  - Instructors, TAs, lab space, ...

Broader Social Issues

- “Something old, something new, …”
- “You’re stealing our courses.”
- “It’s an insidious plot by engineering to take over computer science.”
- “You’re creating a new department.”
- “Software engineering is just a trendy catch phrase, like neural networks.”

Broader Social Issues

- “Those computer scientists don’t understand engineering!”
- “Those engineers don’t understand software!”
- “The students won’t learn enough theory / numerical / AI / hardware / embedded / circuits …”

Issues to be Resolved

- Teaching load, hiring, promotion and tenure
- Support staff, office space
- Techies, lab space
- Fees, BIU levels, low-level academic policies

Where do we go from here?

- Objective: Admit students into the SE program in Sept. 2001
- The hurdles:
  - departmental curriculum committees
  - departments (June 1999)
  - faculty undergraduate studies committees
  - faculty councils (Sept. 1999)
  - senate undergraduate studies committee
  - senate (Oct. 1999)