Software Enterprise Pedagogy for Project Courses

Kevin Gary, Srividya Bansal, Arbi Ghazarian
Department of Engineering
College of Technology & Innovation
Arizona State University
{kgary, sriviyda.bansal, arbi.ghazarian}@asu.edu
Outline

- Project-based learning outcomes
- Software Enterprise delivery model
- Assessing outcomes and industry preparedness
- Professional spine degree program design
- Wrap-up
Project-based learning outcomes
What are the learning outcomes for your project course(s)?

Discuss!
How do you distinguish between the software lifecycle process, progress on project deliverables, and the student learning process?
How do you run projects?
List and prioritize the significant issues

- Industry sponsors
- Teams
- Process
- Just-in-time
- Grading

Project process
Learning process
Outcomes
Software
Enterprise
delivery model
Process-oriented
  – Teams and problems must be large and complex enough to force students to leverage a software process

Sequencing
  – Students are exposed to practices in the order they will progress through them a new professionals in industry

Multi-year
  – The professional spine allows for repetition, sequencing, and variability

Real-world*
  – Exposed to majority of forces affecting industry projects

Collaborative
  – Students learn the key process component is people

We believe these characteristics drive “industry readiness”
If something is a good practice…

…do it all the time!

- scalable
- complex
- mentoring
- problem solving
- real-world
- hands-on
- Good software engineering

project
What you teach is pretty important too.
http://www.ldu.leeds.ac.uk/ldu/sddu_multimedia/kolb/static_version.php
http://goo.gl/jn18N

- Do you do all of them?
- Do you do them all the time?
Software Enterprise Delivery Model

For each Module:
1. Prepare (assess)
2. Lecture (discuss)
3. Practice (competency)
4. Reflect (expectations)
5. Incorporate into project
6. Reflect (internalize)

In a single (3 week) sprint
Software
Enterprise
delivery model

Examples
Software Enterprise Delivery Model

**Keys:**

- Content is commodity
- Learners form expectations
- Co-locate in time
- Project = context
- Variations are very cool
- (force them to) Pause and Reflect
- It is OK that not all projects touch on all content
Assessing Outcomes and Industry Preparedness
Assessment is really hard...

How do you measure *industry preparedness*?

We consider 3 assessment tools:
1. Survey
2. Concept maps
3. [e-] Portfolios / Journals
Survey design

In IEEE Software 2001, Timothy Lethbridge reported on a Canadian national industry survey. He compared what was learned in school versus what one needed to know in the working world. We followed this model.

http://www.computer.org/csdl/mags/co/2000/05/r5044-abs.html
• “For each of the following general skill areas, indicate the level of expertise needed in the area to perform your job successfully.”

**Conclusion**: The longer students are out of school the more they appreciate process and teamwork-related topics
“For each of the specific “Software Engineering” topics below, indicate the level of expertise needed in the area to perform well in your job requirements.”

Conclusion: Students who are fresh-outs tend to be overwhelmed with the technical skillset required.
We chose concept maps as an assessment mechanism for asking if a student’s structural understanding of the body of knowledge is better.
Concept Map design

• Students do pre/post concept maps on a scaffolded map
• Industry alumni asked to do as well
• Experts evaluate the maps for progress

http://lead1.poly.asu.edu/cmap-evaluation
[e-]Portfolio design

“Life can only be understood backwards; but it must be lived forwards.”

— Søren Kierkegaard

We ask students to document reflections and self-assess against program outcomes
[e-]Portfolio design

• Students reflect after practice; they have formed expectations
• Students reflect again after project; they have learned in context and (in)validated expectations
• Beliefs are now internalized

http://swent0.asu.edu/mahara
Professional spine degree program design
We question the work that you do
...er, we mean with regards to projects
When is the last time you ate wedding cake?

...and when is the last time you ate a Twizzler?
### SEEK2004 Recommended General Structure

<table>
<thead>
<tr>
<th>Year 1</th>
<th>Year 2</th>
<th>Year 3</th>
<th>Year 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sem 1A</td>
<td>Sem 1B</td>
<td>Sem 2A</td>
<td>Sem 2B</td>
</tr>
<tr>
<td>CS105</td>
<td>CS106</td>
<td>Calc 1</td>
<td>Calc 2</td>
</tr>
<tr>
<td>NT</td>
<td>SE200/201</td>
<td>SE</td>
<td>SE</td>
</tr>
</tbody>
</table>

#### Intro Computing Sequence

<table>
<thead>
<tr>
<th>Year 3</th>
<th>Year 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sem 3A</td>
<td>Sem 3B</td>
</tr>
<tr>
<td>MA271</td>
<td>SE</td>
</tr>
<tr>
<td>SE</td>
<td>SE</td>
</tr>
</tbody>
</table>

#### More SEEK patterns are available:

http://goo.gl/20zkI

---

### N2S-1c: Year 2 Start with Semesters in a CS department

<table>
<thead>
<tr>
<th>Year 3</th>
<th>Year 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quarter 3A</td>
<td>Quarter 3B</td>
</tr>
<tr>
<td>SE201</td>
<td>SE A</td>
</tr>
<tr>
<td>SE212</td>
<td>SE C</td>
</tr>
<tr>
<td>MA271</td>
<td>Tech elect</td>
</tr>
<tr>
<td>NT272</td>
<td>--</td>
</tr>
</tbody>
</table>

#### N2S-1c: Year 2 Start with Semesters in a CS department

<table>
<thead>
<tr>
<th>Year 3</th>
<th>Year 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quarter 3C</td>
<td>Quarter 4A</td>
</tr>
<tr>
<td>SE D</td>
<td>SE400</td>
</tr>
<tr>
<td>SE E</td>
<td>SE F</td>
</tr>
<tr>
<td>Tech elect</td>
<td>Tech elect</td>
</tr>
</tbody>
</table>

### NSQ-1: Year 3 Start with Quarters

<table>
<thead>
<tr>
<th>Year 1</th>
<th>Year 2</th>
<th>Year 3</th>
<th>Year 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sem 1A</td>
<td>Sem 1B</td>
<td>Sem 2A</td>
<td>Sem 2B</td>
</tr>
<tr>
<td>SE101</td>
<td>SE102</td>
<td>CS103</td>
<td>CS270</td>
</tr>
<tr>
<td>Calc 1</td>
<td>Calc 2</td>
<td>SE200</td>
<td>SE212</td>
</tr>
</tbody>
</table>

#### NSQ-1: Year 3 Start with Quarters

<table>
<thead>
<tr>
<th>Year 3</th>
<th>Year 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quarter 3C</td>
<td>Quarter 4A</td>
</tr>
<tr>
<td>SE270</td>
<td>CS220</td>
</tr>
<tr>
<td>SE A</td>
<td>SE E</td>
</tr>
<tr>
<td>MA271</td>
<td>SE220</td>
</tr>
</tbody>
</table>

#### NSQ-1: Year 3 Start with Quarters

<table>
<thead>
<tr>
<th>Year 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>SE D</td>
</tr>
<tr>
<td>SE F</td>
</tr>
</tbody>
</table>

### N1S: US Model showing starting SE early in CS courses

<table>
<thead>
<tr>
<th>Year 1</th>
<th>Year 2</th>
<th>Year 3</th>
<th>Year 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gen ed</td>
<td>Psychology</td>
<td>NT181</td>
<td>Physics 2</td>
</tr>
</tbody>
</table>

#### N1S: US Model showing starting SE early in CS courses

<table>
<thead>
<tr>
<th>Year 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sci Elect</td>
</tr>
<tr>
<td>NT291</td>
</tr>
</tbody>
</table>

#### N1S: US Model showing starting SE early in CS courses

<table>
<thead>
<tr>
<th>Year 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gen ed</td>
</tr>
</tbody>
</table>

---

Discuss:
Which is close to what you [would] do?

More SEEK patterns are available:

http://goo.gl/20zkI
a project spine provides a better model over the traditional layered approach
1 more thing on outcomes

BS Software Engineering Program Outcomes
Version dated 10/30/2009

Program outcomes identify what students know and are able to do at the time of graduation.

1. **Design.** An ability to design a digital or software system, component, or process to meet desired needs within realistic constraints.
   - Level 1: Recites the steps and information flow in the engineering, digital system and software design process and uses organizational or technical tools in each step.
   - Level 2: Given a problem definition, uses a design process and tools to produce a documented solution design, including a prototype; and explains how the design meets the constraints and criteria.
   - Level 3: Evaluates design process and resulting design quality relative to appropriate criteria, and suggests improvements.
   - Level 4: Customizes design process, related artifacts, and tools for specific problem contexts.

2. **Computing Practice.** An ability to employ knowledge, methods, skills, modern software engineering tools, and team-based approaches appropriate for interdisciplinary and multi-cultural settings in arriving at computing problem solutions.
   - Level 1: Describes the essential elements of team-based computing practice.
   - Level 2: Given a problem statement, works within a team to develop and carry-out a plan that identifies tools and methods to produce a technical solution.
   - Level 3: Evaluates the effectiveness of the planning process, teamwork and tools used to produce a solution to a computing problem.
   - Level 4: Adapts the planning process, team work approach, tool sets, and methods to a problem context, and then professionally applies them to obtain optimal (defensible) solutions to computing problems.

3. **Critical Thinking and Decision Making.** An ability to think critically, clearly identifying and using evidence, criteria, and values in the decision making process.
   - Level 1: Articulates the critical thinking process.
   - Level 2: Identifies assumptions, criteria, and evidence to make informed decisions.
   - Level 4: Examines and cultivates a value system to make informed decisions.

4. **Professionalism.** An understanding of professional and ethical responsibility, a commitment to on-going professional competence and possession of basic professional and organizational success skills.
   - Level 1: Identifies professionally appropriate behavior. Appreciates engineering and computing as a learned profession, and possesses daily success skills.
   - Level 2: Accepts responsibility for personal education. Understands the major professional and ethical responsibilities of computing professionals. Understands the major computing specialties, corporate structures and purposes.
   - Level 3: Uses accepted ethics and concepts to guide decisions, and formulates a career path that accounts for current trends in technology and society.
   - Level 4: Effectively guide activities to maintain professional competence and reputation.

5. **Perspective.** An understanding of the role and impact of engineering and computing technology in business, global, economic, environmental, and societal contexts.
   - Level 1: Understands that technological change and development have both positive and negative effects.
   - Level 2: Identifies and evaluates the assumptions made by others in their description of the role and impact of engineering and computing on the world.
   - Level 3: Selects from different scenarios for the future and appropriately adapts them to match current technical, social, economic and political concerns.
   - Level 4: Has formed a constructive model for the future of our society, and makes life and career decisions that are influenced by the model.

6. **Problem Solving.** An ability to identify, formulate, and solve problems whose solutions require digital and software systems using the steps of an analytical problem solving approach.
   - Level 1: Articulates the problem solving process by making explicit the steps taken to realize a solution.
   - Level 2: Performs all steps of the problem solving process including conceptualization, elaboration, realization and analysis in both closed and open-ended design and analysis problems.
   - Level 3: Analyzes, selects, uses, and evaluates various methods and frameworks for developing solutions to computing problems.
   - Level 4: Adapts tools, methods and frameworks of problem solving to a variety of computing problems requiring a collaborative teamwork approach.

7. **Communication.** An ability to communicate effectively.
   - Level 1: Recognizes and describes individual processes used in various modes of communication.
   - Level 2: Uses a process to develop appropriately structured communications.
   - Level 3: Purposefully applies communication strategies to interact meaningfully with the audience.
   - Level 4: Selects and adapts communication strategies to fully engage the audience.

8. **Technical Competence.** An ability to apply, evaluate and adapt knowledge of mathematics, software, computer science, computer systems, and software engineering in generating complex solutions to problems in a software engineering application area.
   - Level 1: Verbally and mathematically communicates the fundamental software, computer science, computer systems and software engineering principles, methods, and tools underlying computing solutions.
   - Level 2: Applies fundamental computing and software engineering principles, methods, and tools to construct solutions to computing problems using a defined approach including analysis, modeling, prototyping, and validation.
   - Level 3: Evaluates, selects and applies appropriate fundamental computing and software engineering principles, methods, and tools relevant to characteristics of a software engineering application area in a structured manner.
   - Level 4: Selects, adapts, and applies appropriate fundamental computing and software engineering principles, methods and tools to solve complex multidisciplinary computing problems.
Wrap-up
Software Enterprise resources

~40 curricular modules available
Assessment software and practices
Publications and advice
Forming a community of adopters

http://softwareenterprise.asu.edu
Learn by doing
Learn in context
{ prepare
  assess
  discuss
  describe
  practice
  expect
  project
  context }
The Software Enterprise
http://softwareenterprise.asu.edu