The Computer Science Professionals (CSP) Hatchery will create a revolutionary learning environment by modeling the best practices of a software company work experience, layering nurturing aspects that promote ethical questioning, value diversity, and a focus on professional skills such as increased collaboration, communication, and teamwork.

**Abstract**
The Computer Science Professionals Hatchery seeks to transform undergraduate education by replicating the best elements of a software company environment, laying in moral, ethical, and social threads with entrepreneurship and professional skills. The Computer Science Professionals Hatchery focuses on three curricular innovations: (1) VERTICAL INTEGRATION. Instead of being siloed, students at all grade levels will work with and learn from each other in industry-sponsored projects, fostering a strong sense of community amongst students, faculty, and industry. (2) Short, narrowly focused HATCHERY UNITS will complement regular course work by presenting aspects of specific, foundational concepts or skills that cut across the curriculum, using a unique approach that overlays agile Hatchery Units with regular courses. (3) ETHICS AND SOCIAL JUSTICE will be incorporated across the curriculum to encourage students’ development as professionals and empower them to be agents of change in reshaping computer science to be a more just and inclusive profession.

**Theoretical Framework**
- Rogers’ (2003) theory of diffusion of innovation guides our approach to creating lasting change in our department.
- Designing a vertically integrated curriculum that builds connections across grade levels is informed by Wenger’s (1998) Communities of Practice as we build community through establishing norms and developed shared understanding.
- Our approach to incorporating ethics and social justice in the computer science curriculum is grounded in Rawls’ (1999) theory of social justice.

**Guiding Questions**
1. How can ethics and social justice be incorporated into an undergraduate computer science curriculum?
2. How does the transformed curriculum influence students’ undergraduate experience?
3. What are the barriers and supports to curriculum change and beyond?

**Objectives**
- Create a Culture of Engagement
  - Develop faculty members who understand and are invested in the major elements of the curriculum across all 4 years.
  - Build a strong sense of community amongst faculty, students, and industry partners, with stronger ties and working relationships between these constituencies.
  - Overcome barriers and implement sustainable drivers for curricular innovation.

- VITaL Vertically Integrated Teaching and Learning
  - Improve real-world relevance of the student educational experience, leading to improved student motivations and job preparation.
  - Fuse professional and entrepreneurial skills learning throughout the curriculum that complements technical skills learning.

- Create a Diversity-Promoting Revolution
  - Implement a progressive introduction of best practices supporting diversity and ethical/moral professional practice to sensitize students.
  - Develop and teach approaches for incoercibly infusing ethical/moral elements into the practice of software engineering to train students to develop agents of change in their workplaces.

**Hatchery Change Process**
Hatchery Units (HUs) are one credit courses focused on skills relevant to computer science professionals and designed to rapidly adapt to the changing needs of industry. HUs are also a vehicle to diffuse social justice and equity through the curriculum.

**Hatchery Curriculum Map**
- CS-HU 130: Foundations Values
- CS-HU 271: Agile Development
- CS-HU 350: Intro to Database System Design
- CS-HU 375: Secure Programming

**Hatchery Units**

<table>
<thead>
<tr>
<th>HU Course</th>
<th>Start</th>
<th># Students</th>
</tr>
</thead>
<tbody>
<tr>
<td>Foundations Values (CS-HU 130)</td>
<td>Fa'17</td>
<td>232</td>
</tr>
<tr>
<td>Agile Development (CS-HU 271)</td>
<td>Fa'17</td>
<td>52</td>
</tr>
<tr>
<td>Navigating Computer Systems (CS-HU 350)</td>
<td>Sp'18</td>
<td>336</td>
</tr>
<tr>
<td>Intro to Database System Design (CS-HU 375)</td>
<td>Sp'18</td>
<td>42</td>
</tr>
<tr>
<td>Intro to Version Control (CS-HU 350)</td>
<td>Sp'18</td>
<td>15</td>
</tr>
<tr>
<td>Technical Interviews, Jobs &amp; Careers (CS-HU 390)</td>
<td>Fa'17</td>
<td>14</td>
</tr>
<tr>
<td>Brief Intro to HCI (CS-HU 269)</td>
<td>Fa'18</td>
<td>--</td>
</tr>
<tr>
<td>Current Topics in Computer Science (CS-HU 268)</td>
<td>Sp'18</td>
<td>--</td>
</tr>
<tr>
<td>Software Testing (CS-HU 375)</td>
<td>Fa'18</td>
<td>--</td>
</tr>
<tr>
<td>Secure Programming (CS-HU 375)</td>
<td>Fa'18</td>
<td>--</td>
</tr>
</tbody>
</table>

**References**