

Disclaimer: Please note that the specifics of this Course Syllabus can be changed at any time, and you will be responsible for abiding by any such changes. All changes will be communicated with you via Blackboard course announcement and/or update on Blackboard's syllabus section.

**SWEN 5234**  
**Software Processes**  
**Fall 2016**

**1. General information**

**1.1. Class details**

SWEN 5234.01 Class Number 10466 Wednesday 1.00PM – 3.50pm Delta 140

**1.2. Instructor**

Soma Datta, Ph.D. Assistant Professor Software Engineering  
Delta 225, 281-283-3838, [datta@uhcl.edu](mailto:datta@uhcl.edu)  
Office hours: 11:00 -12:50 Tu/W/Th

**1.3. Teaching Assistant**

Kenneth Akpo  
Office Hours: Tu 9 AM to 12 PM, W 10 AM to 3 PM, Th 4 PM to 8 PM

**1.4. UHCL Quality Enhancement Plan (QEP) Motto: Applied Critical Thinking (ACT) for Lifelong Learning and Adaptability**



**This course emphasizes Communication and Concepts**

This course has been authorized by UHCL as an Applied Critical Thinking (ACT) Course which means that in addition to learning about the specified course content, students will be engaged with some or all of the Elements of Thought and Universal Intellectual Standards of critical thinking. The objective of an ACT course is to develop the student's ability to become skilled at analysis and evaluation by applying a set of intellectual tools that may be effectively used across all disciplines (as well as to the student's personal life). Based on the Foundation for Critical Thinking model (<http://www.criticalthinking.org/>), critical thinking involves thinking for a *purpose*, asking *questions*, using *information*, applying *concepts*, drawing *inferences and conclusions*, identifying *assumptions*, anticipating *implications and consequences*, and recognizing *points of view*. The Universal Intellectual Standards that are applied to these Elements of Thought of critical thinking in order to develop Intellectual Traits include *clarity*, *accuracy*, *precision*, *relevance*, *depth*, *breadth*, *logic*, *significance*, and *fairness*.

**Critical Thinking in Software Engineering in General and Software Processes in Particular**

Software engineering is the application of engineering to the design, development, implementation, testing and maintenance of software in a systematic method.

Typical formal definitions of software engineering are:

- Research, design, develop, and test operating systems-level software, compilers, and network distribution software for medical, industrial, military,

communications, aerospace, business, scientific, and general computing applications.

- The systematic application of scientific and technological knowledge, methods, and experience to the design, implementation, testing, and documentation of software.
- The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software.
- An engineering discipline that is concerned with all aspects of software production and the establishment and use of sound engineering principles in order to economically obtain software that is reliable and works efficiently on real machines (from Wikipedia).

In software engineering, a software processes (also known as a system development methodology, software development life cycle, software development process) is a splitting of software development work into distinct phases (or stages) containing activities with the intent of better planning and management. It is often considered a subset of the systems development life cycle. The methodology may include the pre-definition of specific deliverables and artifacts that are created and completed by a project team to develop or maintain an application.

Common methodologies include waterfall, prototyping, iterative and incremental development, spiral development, rapid application development, extreme programming and various types of agile methodology. Some people consider a life-cycle "model" a more general term for a category of methodologies and a software development "process" a more specific term to refer to a specific process chosen by a specific organization. For example, there are many specific software development processes that fit the spiral life-cycle model (from Wikipedia).

Critical thinking helps in clearly understanding the concepts and relevance of different software processes. After this course, we will be able to explain the significance, implications and consequences associated with selecting a software process. We will be able to clearly identify the information necessary for a given software process and explain its relevance to the identified problem.

### **1.5. Fundamental and Powerful Concepts**

This course includes three important concepts that are essential to understanding and applying software processes.

1. Significant issues associated with selecting and applying a software process to a project using the six best practices.
2. The six best practices are:
  1. Develop software iteratively
  2. Manage requirements
  3. Use an architecture that helps in reuse and reengineer a software
  4. Visually model software
  5. Verify software quality
  6. Control changes to software
3. The Basic methods or software processes are:

Waterfall, prototyping, iterative and incremental development, spiral development, rapid application development, extreme programming and various types of agile methodology.

## 1.6. Course Description

From Catalog: Detailed coverage of the theory, application, assessment and evaluation of the Unified Process Model. Course will cover the process modeling, process assessment, quality assessment of process models and process improvement techniques. Introduction to Agile and other processes.

The course introduces several software processes through real life examples and research papers.

## 1.7. Course Structure

This class is problem solving and discussion orientated. First part of the course will be concept presentation; it will be followed by examples about different processes. The class will be divided into groups and each group will be assigned projects/processes and papers to lead the discussions. The final part of this course is the preparation for in-class presentation of a publication-quality research paper.

## 1.8. Student Learning Outcomes (SLO)

After completion of the course the students are expected to be able to:

- SLO 1. Understand the *concepts* and *relevance* of different software processes
- SLO 2. Understand the *significant implications and consequences* associated with selecting a software process
- SLO 3. Clearly identify the *information* and explain its relevance to the problem
- SLO 4. *Clearly* identify the *problem* and explain the *purpose* of a software process
- SLO 5. Understand and *identify* the complexity (*depth*) of a software process to be used in the project
- SLO 6. Discuss the *breadth* of a given problem by finding an *alternative* path and explain different software processes
- SLO 7. Understand the responsibilities and *relevance* of the core roles of the individuals (*depth*) in a software process and be able to assign it to staff/employees such as: project managers, requirements managers, architects, designers, developers, test managers, testers, quality managers, change agents, and finally customers.
- SLO 8. Understand the execution (*purpose, problem, depth, breadth, and relevance*) of the core elements in a software process and be able to demonstrate; such as: requirements elicitation and management, project management, design management, development, test management, testing, quality control (reviewing), deliveries and deliverables, and finally customer involvement.

Note: The italicized word(s) are related to critical thinking. Please refer to the section ACT.

## 1.9. Applied Critical Thinking (ACT)

### Vocabulary of Critical Thinking

We use the vocabulary of critical thinking described by Drs. Richard Paul and Linda Elder, including the eight elements of thought and nine universal intellectual standards:

Eight elements of Thought of Critical Thinking:

1. *Purpose*
2. *Question at Issue*
3. *Information*
4. *Interpretation and Inference*
5. *Concepts*
6. *Assumptions*
7. *Implications and Consequences*
8. *Point of View*

Nine Universal Intellectual Standards for Critical Thinking:

1. ***Clarity***
2. ***Accuracy***
3. ***Precision***
4. ***Relevance***
5. ***Depth***
6. ***Breadth***
7. ***Logic***
8. ***Significance***
9. ***Fairness***

For more details, see:

[1] Paul, R. and Linda Elder, L., *The Miniature Guide to Critical Thinking Concepts and Tools (Thinker's Guide)*, 7th edition, Foundation for Critical Thinking, 2014.

[2] Paul, R. and Linda Elder, L., *Critical Thinking: Tools for Taking Charge of Your Learning and Your Life*, 3rd Edition, Prentice Hall, 2011.

#### Critical Thinking Process (CTP)

According to the ACT vocabulary we used, there are four major aspects of the Applied Critical Thinking Process, termed as the 4 C's: curiosity, connections, creativity and communication. In this course, the C in the student learning objectives is connections: Making connections to a particular issue or problem: students will use established academic and industrial software processes to model a problem, design and implement a software development process, and be able to find alternatives to the process.

#### Critical Thinking Activities and Assessment

There are two assessment activities (AA) of critical thinking in the course. The evaluation of these activities is used to assess how well critical thinking is incorporated into the course. These assessments will be used as input to the UHCL Critical Thinking software processes for internal assessment of Critical Thinking, and will not affect your grade of the course. The details of this assessment is mentioned in section 2.3.

## **2. Course Grading and Deliverables**

Attendance Mandatory

### **2.1. Class Participation (8%)**

Students are expected to read about the topic in a Class, come to class prepared to discuss

their thoughts and take part of the classroom discussions. Each class will add 1% towards your class participation.

## 2.2. Short written exercise/Software Processes/in class work (30%)

Each team will write/detail the software process used and upload as per the assignment date by 11.59 PM using the following template:

- A paragraph (3-5 sentences) about the process
- At least two to three bullet points to highlight the advantages of the process they used.
- At least two bullet point of the disadvantages of the process used
- Mention if applicable for an alternative process and why

A template is uploaded on Blackboard.

## 2.3. Term paper (25% document+12% presentation) SLO 1, SLO 2, SLO 3, and SLO 4

A large portion of the course deliverables is a course term project. You are expected to work on the course project in groups of 2-3 students. Each group is expected to write a term paper by the end of the semester. The topic is to be discussed with the instructor. Examples include specific software processes such as Rational Unified Process, Kanban, Waterfall, Spiral (typically involves at least 20 pages), or building a software using one of the software processes with detailed discussion on comparing different software processes.

The final report (minimum 20 pages in length) should be submitted by the end of Class 12. The final submission is expected to be of detailed and quality work as per the example given on blackboard. Please refer to the rubrics on “Term Paper” for detailed assessment of these SLOs

- SLO 1. In your term paper you will choice a specific software processes and **clearly** explain the *concepts* of that process. You will then explain the **relevance** and why you need to use that software process.
- SLO 2. In your term paper you will discuss each steps of the process and how you are going to implement that process, you will give an example and explain the **significant** implications and *consequences* why and how the process you selected suits best.
- SLO 3. You will conclude your term paper by summarizing the information received and its **relevance** to the given *problem*.
- SLO 4. You will **clearly** explain the *purpose* and its **relevance** to the software process you have selected.

2.4. Exam 1 15%

2.5. Exam 2 10%

## 3. Grading Policies

### Grading Assignment Table

<b>Letter Grade</b>	<b>Grade Points Per Semester Hour</b>	<b>Grade Points</b>
A	4.00	93.0-100.0
A-	3.667	90.0-92.9
B+	3.333	87.1-89.9
B	3.000	85.0-87.0
B-	2.667	80.0-84.9
C+	2.333	77.1-79.9
C	2.000	73.0-77.0
C-	1.667	70.0-72.9
D+	1.333	67.1-69.9
D	1.000	63.0-67.0
D-	0.667	60.0-62.9
F	0.000	below 60.0

#### 4. Course Schedule

<b>Class</b>	<b>Session</b>	<b>Class Activities</b>
<b>Class Week 1</b>	<ul style="list-style-type: none"> <li>• Introduction/ Syllabus</li> </ul>	Decide on groups
<b>Class Week 2</b>	<ul style="list-style-type: none"> <li>• What is Software process? Software Lifecycle, purpose, knowledge of processes</li> </ul>	Class Discussions (CD)
<b>Class Week 3</b>	<ul style="list-style-type: none"> <li>• Waterfall, V model, Spiral model, Rational Unified Process, Agile Process</li> </ul>	CD
<b>Class Week 4</b>	<ul style="list-style-type: none"> <li>• Project Discussions</li> </ul>	Group Discussions (GD)
<b>Class Week 5</b>	<ul style="list-style-type: none"> <li>• Waterfall, V model, Spiral model, Rational Unified Process, Agile Processes/ Paper presentations</li> </ul>	CD, 30 mins GD, 5 mins. group presentation. **
<b>Class Week 6</b>	<ul style="list-style-type: none"> <li>• Waterfall, V model, Spiral model, Rational Unified Process, Agile Processes/ Paper presentations</li> </ul>	CD, 30 mins GD, 5 mins. group presentation. **
<b>Class Week 7</b>	<ul style="list-style-type: none"> <li>• Mid-Term</li> </ul>	Individual Test taking
<b>Class Week 8</b>	<ul style="list-style-type: none"> <li>• Project Prep day</li> </ul>	Teams work in class
<b>Class Week 9</b>	<ul style="list-style-type: none"> <li>• Core actors in software processes/ DFD's</li> </ul>	**

<b>Class</b>	<b>Session</b>	<b>Class Activities</b>
<b>Class Week 10</b>	• DFD's	
<b>Class Week 11</b>	• DFD's	Debate in Class
<b>Class Week 12</b>	• Project Presentations	Group Presentations
<b>Class Week 13</b>	• Project Presentations	Group Presentations
<b>Class Week 14</b>	• Review	
<b>Class Week 15</b>	• Finals	Individual Test taking

## **UHCL Policies**

### **Honesty Code**

The Honesty Code is the university community's standard of honesty and is endorsed by all members of the University of Houston-Clear Lake academic community. It is an essential element of the University's academic credibility.

It states:

I will be honest in all my academic activities and will not tolerate dishonesty.

### **Disabilities**

If you have any special needs due to a disability, please let me know. For information on disability accommodations and access, please contact the Disability Services Office, Bayou Room 1402 or call 281-283-2627.