

Syllabus - CSI 3520 Systems Analysis
EC275

Tuesday-Thursday 0800-0947
First Class: Thursday, January 4th, 2018

Leon Preston Brooks, Jr. (Preston)
Adjunct Associate Professor
Computer Science and Engineering Department
Oakland University
brooks2@oakland.edu
Cell: (248) 535-2665

Office Hours: Thursdays, 0700-0800 in the classroom and by appointment

Course Objective 1.: Understand the Overall Information Technology System Development Process.

This course is different from the tool or programming language-specific courses you have taken so far. In this course, we as engineers will have the opportunity to solve a business problem. Remember that all software development is a result of the need to solve a business problem in industry. What we don't get enough exposure to is the art and science of taking information from a customer and from this information, crafting a clearly stated business problem. From this business problem statement, we will extract a set of solution requirements that will lead us to a proposal for a software application operational concept that will satisfy the solution requirements and solve the business problem. That is, we will focus on "the front end of the problem", the part that we software engineers rarely get an opportunity to participate in.

In most of our technical courses, we start with the software operational concept, extract the system-level requirements, develop the UML model and simultaneously develop the Java code using test-driven development (TDD) with JUnit. In this course, our objective is to define a business problem as described by the customer and translate it to a software application solution concept that will solve the customer's business problem. The goal of the course is to develop (1) a software application solution operational concept, (2) a requirements set and (3) a UML model that will solve the customer's business problem.

Course Objective 2.: Apply Software Engineering Principles to Software Development.

The foundations of this course are (1) translating a business problem into a software solution operational concept and a set of software application solution requirements, (2) a Unified Modeling Language (UML) modeling of requirements, (2) apply Agile Java software engineering principles and (3) understand Test-Driven Development (TDD) using JUnit. The concept of discovering software defects early in development using pair programming will be presented. The course will stress the concept of "continuous code review" by pair Java programming partners ("Xtreme Programming"). There will

be no Java programming in this course. We will, however, develop a software solution model using UML in the free UML modeling tool, Astah.

Course Objective 3 : Understand the Requirements Gathering and Validation Process.

Understanding, analyzing and developing the requirements for an application is the foundation of good Agile Java Software Development. It is crucial for the project team to work closely with the Customer and his/her representatives to ensure that the Operation Concept for the software solution application will meet the Customer's business needs. During this course, the Instructor will act as the Customer for project purposes. Once the software solution application Operational Concept is defined and agreed, the team extracts the system-level requirements ("shall statements") from the Operational Concept. The extracted set of system-level requirements must be validated by the Customer. We will then create a UML model of these system-level requirements.

Course Objective 4 : Understand the basics of Project Management using MS Project 2016

We will study the basics of how to plan a software development project using MS Project 2016 as part of the systems analysis process. We will cover construction of a Work Breakdown Structure (WBS) and Gantt chart for a proposed project. We will discuss how the project is structured and how to build a plan using categories of tasks, called "workflows. We will also discuss the concept of a task and subtask, along with task durations and dependencies as represented in an MS Project Gantt chart. Additionally, we will discuss risk in a project and how to quantify it and manage it.

Root Cause Analysis

We will use the Root Cause Analysis (RCA) process to analyze the business cases. Once we have the "root cause" of all the symptoms, we will develop an operational concept for the solution, including a software App. Once we have the operational concept we will extract the system level requirements and construct a UML Model of the App.

Course Administration and Grading

Please see the "**Course Policy and Procedure**" posted on Moodle.

Class Project

I will divide the class into teams, each with a team lead. There will also be a Lead Systems Integrator (1) team headed by the project manager for the project. There will be subsystem teams responsible for (2) Display, (3) Middleware and (4) Database. The class project will deal with analyzing some business problem and developing an Operational Concept for a software application to solve this problem. We will use all the tools we have learned in this course in the development and execution of the class project.

Software Tools

Astah UML Modeling Tool

You will be learning the basics of UML modeling using the Astah UML Modeling tool. The community edition of this free UML tool can be obtained from the website below. Please download and install Astah prior to the first class. UML will be used extensively in our discussions of software application development using the Agile process.

<http://astah.net/editions/community>

MS Project 2016

You should download MS Project 2016 from the OU SECS software website, http://www.oakland.edu/secs/cto/software_distribution . We will build a class project plan (Gantt chart) as part of our systems analysis. Please download and install MS Project 2016 prior to the first class.

Textbooks

The *recommended* textbook for the course is the following text, *“Systems Analysis and Design with UML”*, 5th Edition, Dennis, Wixom and Tegarden, Wiley, 2012, ISBN 978-1-118-05762-9). This book should be available on line at www.mysafaribooks.com and at Amazon and the usual suspects. It will be advantageous for you to have access to this book to take advantage of the lectures and the Minicases.

References

I intend to use the following texts as well as Instructor's Notes as references for the course. All can be found on www.my.safaribooks.com . The site has over 46,000 technical books and videos. You can get a trial membership for at no cost.

Additional Helpful References:

“UML Distilled”, 3rd Edition, Martin Fowler, 2003, Print ISBN-13: 978-0-321-19368-1

“Microsoft Project 2013, in Depth”, Scott Daley, Que Publications, ISBN 13: 978-0-7897-5095-2

“Agile Java: Crafting Code with Test-Driven Development,” Jeff Langr, 2006, Prentice-Hall, ISBN-13: 978-0-13-148239-5

“Extreme Programming Explained: Embrace Change”, 2nd Edition, Kent Beck, 2004, Print ISBN-13: 978-0-321-27865-4

Please consult the Class Schedule on Moodle for the detailed assignments (reading and homework) for each class. You are expected to be prepared for each class.