

**CIS 641/642 – Software Engineering Design Project  
Fall 2016**

**Instructor:** Nathan Bean

**Email:** nhbean@ksu.edu

**Class Time & Location:** Monday, Wednesday, & Friday 9:30pm-11:20pm, DUE 1116

**Office Hours & Location:** Tuesday, 3:30pm-4:30pm or by appointment (<https://schedule.cis.ksu.edu>), DUE 2216

**Final Exam Period:** 11:50am – 1:40pm, Thursday, December 15

**GTA:** Adedolapo Okanlawon

**Office Hours & Location:** Monday 2:30pm-3:30pm, DUE 1119

### **Course Description**

From the K-State Course catalog:

Current Practices of software development, requirements, design, prototyping, measures, and evaluations. Specification, design, and prototyping of a software system.

In plain English:

You will be creating commercial software to tackle a real-world application that will be released as an open-source project to the public. This includes:

- *Starting from an existing code base.* Very rarely will you have the luxury of starting a real-world project from scratch. Almost always, you'll have to start from an existing program, framework, or API specification, and adapt it to your customers' needs. Existing code bases are often poorly or incorrectly documented, under- or completely non-tested, and full of spaghetti code.
- *Working as a team.* You will very likely never work on a solo project in industry - modern software engineering involves teams of tens if not hundreds of programmers, plus many non-technical contributors as well. Good software development therefore involves tools and strategies to minimize code collisions and endure productive and healthy working relationships.
- *Working directly with customers* to understand what the software needs to do and how it can be most useful to them. This includes usability design and testing, feature negotiation, and human relations.
- *Documenting* the software as you develop it in a way that supports future maintenance and feature additions.
- Creating *test frameworks* or applying *validation frameworks* to ensure your code does exactly what it is intended to do without *any* errors.
- *Refactoring* spaghetti code and poor designs for greater efficiency and maintainability.
- Using *prototypes* to explore new feature ideas for usability and functionality.
- Employing a *staged design, testing, and deployment process* to ensure that only the highest-quality, functional, and well-tested code ever appears in front of regular users.

This bulleted list captures the course objectives - by the completion of the course you will need to have demonstrated mastery of each point through developing your software engineering project to earn an A.

### **Course Structure**

This course is primarily a project course – you will be grouped into teams by the instructor and assigned a specific, real-world software application to develop. Each project also comes with realworld customers who will provide guidance and design priorities, judge usability, negotiate development schedules, and determine how well your software meets their needs. Your projects will be released as open-source projects and hosted on a public Github repository. You should realize that this project will be one of the criteria by which your future employers will judge your suitability, as well as the recommendations of both your instructor and customers.

### **Lectures, Readings, and Class Discussions**

While the core focus of this course is *learning by doing*, carrying out the principles of good software engineering while creating or expanding upon an open-source software project to meet society's' needs, understanding the process begins with studying engineering practices.

### **Software Projects**

Software projects will cover a wide range of uses as well as base technologies/development languages. The instructor will endeavor to match you to a project in your interest area, but this will not always be possible. In all cases, these are real-world projects that will go on to be used by hundreds or thousands of real-world users. Good design is essential and serves to support both these eventual users and to build your reputation as a software engineer.

Software project development will proceed using an Agile methodology with a two-week iterative development cycle. At the end of each cycle, you will release a tagged prototype version of your software using Git's tagging functionality. This prototype will be graded for: 1) code quality – focusing on readability, maintainability, and efficiency, 2) documentation – focused on accuracy and coverage, 3) testing/verification processes – focused on depth and correctness of coverage.

You will conduct a stand-up meeting, a sprint review meeting, or a sprint planning meeting (or any combination of the above) during the assigned class period. As such, it is vital to the success of your project that you attend these meetings! This will be the basis of your class participation grade.

### **Development Teams**

You will be assigned to a development team of 3-4 students. Each student is expected to contribute to all aspects of the development process. This means that you will not have a documentation guru or a testing guru who only writes documentation or tests. Every member should carry part of the responsibility for writing code, tests, and documentation, as well as working with customers. However, you can divide duties in such a way for a specific design iteration – but by the end of the project each member should have contributed equally in each area.

Every two weeks you will have a formal review and planning meeting with some or all of your customers, at which point they will evaluate your iteration's prototype for how well it meets their needs. This meeting will be scheduled during the course lab time to facilitate coordination of schedules. It is a mark of respect and professionalism to be on-time, well-prepared, and well groomed for these meetings. Missing a meeting (with the exception of emergency circumstances) will result in an automatic lowering of your grade by a full letter

(1 on a 4-point scale). The customers' evaluations, along with a survey of how well they felt the team interacted with them during the design iteration, will be the basis of your customer relation grade. This grade will be adjusted by the instructors if the team members arrive late, are ill-prepared, appear in inappropriate attire (for these meetings, you should be in formal business wear).

**Appropriate Tool and Artifact Usage**

Each team will also be responsible for making adequate usage of the many tools developed to enhance and expedite the software development process. At a minimum, the development team should be using 1) a code repository to manage the sharing of team code (Git publicly hosted through GitHub is required) 2) a feature planning and bug tracking system (Trello is required), and 3) a documentation wiki (Use the Github built-in Wiki). Each of these is a tool you should be using throughout the development cycle, not simply something you throw data into right before your next meeting. You will be graded on how regularly and effectively you utilize these tools.

In addition, Agile software development utilizes a set of artifacts – physical and visual representations of the project's goals and progress. These help teams and managers quickly assess what needs to be done on a project and how far along the process is. Your team will also need to make appropriate and regular use of these artifacts, including: 1) velocity chart, and 2) burn-down chart.

**Grading**

Grades will be issued on a 4-point scale, with 4=A and 0=F. Grades will be issued for each category according to this breakdown:

Grade Category	Consists Of	% Final Grade
Software Projects	Code Reviews, Documentation, Testing, Functionality	25%
Development Teams	Peer Reviews, Meeting Attendance, Project Contributions	25%
Customer Relations	Usability of Application, Quality of Interactions, Professionalism	25%
Appropriate Tool Usage	Regularity of usage, appropriate and full usage	25%

In addition, your overall course grade cannot exceed the score you've earned in any one category, ie. if you earned a 3.8, 4.0, 3.4, 3.2, and 2.1, your final grade would be clamped at 2.1 (C letter grade). This is because each category is equally important for your preparation for becoming a software engineer - remember, it isn't only about writing good code, it's about making that code useful to society!

**Other Course Policy Statements**

**Attendance Policy**

As laid out in the individual course structure subsections, missing a team or customer meeting will result in the reduction of your final grade in the corresponding category by a full letter grade. A further absence will lower your grade another letter. Further, in accordance with the grade policy, this would also lower your potential final grade by the same amount.

### **Late Work**

Keeping to a set development cycle is a crucial aspect of real-world development. Accordingly, there will be no late work accepted.

### **Dress Policy**

As indicated in the customer relations section, students are expected to wear formal business attire for the bi-weekly review and planning meetings with their customers. For other meetings with customers, business casual will be appropriate. Career and Employment Services publishes guides to professional and casual attire for men and women on their site (<http://www.kstate.edu/ces/students/dresstoimpress.html>). These will be used as the grading rubric within this area, so be sure to familiarize yourself with the expectations!

For obtaining inexpensive business formal and casual wear options, there are many opportunities available to you:

- CES offers the *career closet* every Monday-Thursday, Noon - 4pm at the Berney Family Welcome center to find gently-used professional wear. Students can typically take up to five articles. Monetary and canned food donations will go to the Manhattan Emergency Shelter.
- Several thrift stores in Manhattan carry business professional and casual attire at affordable prices (listed by proximity to campus):
  - Rockstar and Rogers Home Fusion, 316 Poyntz Avenue
  - Salvation Army, 310 Poyntz Avenue
  - Goodwill, 421 E. Poyntz Avenue
  - Grand Old Trunk, 1304 Pillsbury Drive

### **Subject to Change**

The details in this document are not set in stone – there may need to be adjustments made throughout the semester. If this occurs, changes will be posted to the K-State online page and emailed to student's K-State email addresses.

### **Academic Honesty**

Kansas State University has an Honor System based on personal integrity, which is presumed to be sufficient assurance that, in academic matters, one's work is performed honestly and without unauthorized assistance. Undergraduate and graduate students, by registration, acknowledge the jurisdiction of the Honor System. The policies and procedures of the Honor System apply to all full and part-time students enrolled in undergraduate and graduate courses on-campus, off-campus, and via distance learning. The honor system website can be reached via the following URL: [www.ksu.edu/honor](http://www.ksu.edu/honor).

A component vital to the Honor System is the inclusion of the Honor Pledge which applies to all assignments, examinations, or other course work undertaken by students. The Honor Pledge is implied, whether or not it is stated: "On my honor, as a student, I have neither given nor received unauthorized aid on this academic work." A grade of XF can result from a breach of academic honesty. The F indicates failure in the course; the X indicates the reason is an Honor Pledge violation.

In the realm of software development, this extends to using code that you have not written without express permission (typically issued through a license agreement). If you use licensed code in your projects, you must abide by the terms of that license, including providing a copy of the license and clearly identifying which parts of the project that includes. Failure to do so, or incorporating code that you do not have a legal

right to use, will be considered an act of plagiarism and submitted to the honor council as a violation of the honor pledge.

### ***Students with Disabilities***

"Any student with a disability who needs a classroom accommodation, access to technology or other academic assistance in this course should contact Disability Support Services ([dss@kstate.edu](mailto:dss@kstate.edu)) and/or the instructor. DSS serves students with a wide range of disabilities including, but not limited to, physical disabilities, sensory impairments, learning disabilities, attention deficit disorder, depression, and anxiety."

### ***Expectations for Classroom Conduct***

All student activities in the University, including this course, are governed by the Student Judicial Conduct Code as outlined in the Student Government Association By Laws, Article VI, Section 3, number 2. Students that engage in behavior that disrupts the learning environment may be asked to leave the class.

### ***Campus Safety***

Kansas State University is committed to providing a safe teaching and learning environment for student and faculty members. In order to enhance your safety in the unlikely case of a campus emergency make sure that you know where and how to quickly exit your classroom and how to follow any emergency directives. To view additional campus emergency information go to the University's main page, [www.k-state.edu](http://www.k-state.edu), and click on the Emergency Information button.