ISAT 345--Software Engineering--Syllabus

Fall 2016

Course Overview

Collaboration is the hardest part!!!

What is this course about?

ISAT 345 is designed to teach you the tools and processes that allow medium to large teams to design and build software. Software development is hard enough with two or three people, but it becomes exponentially more difficult to manage the development process the more people you add to the project. In this class you will learn by doing–you will learn software engineering methodology by being part of a large software development team.

What methodology will we use?

There are many different methodologies for software engineering. This class will focus on a subset of those methodologies referred to collectively as “agile.” While the label “agile” was only coined in 2001, agile methodologies have their roots in “lightweight” software development processes invented in the 1990’s, and have grown tremendously in popularity over the past twenty years. As you enter the workforce, you will find that agile practices have now become standard in most domains of software development. Knowing them prior to being hired will be a **major** selling point to potential employers.

More specifically, we will be focusing on two agile methodologies in particular: scrum and kanban. Scrum and kanban are widely used by companies like IBM and Accenture, and so having experience with these methodologies will improve your chances of employment and ease your transition into a professional software development shop.

What software will we build?

This semester we will be working on building a system called SmartClickr. SmartClickr is a combination web-based and mobile audience response system (ARS) similar to the “clickers” you may have used in a classroom setting. SmartClickr replaces the dedicated hardware of clickers with smartphones, tablets, computers, and the internet. You’ll learn much more about the project as we move forward.

Our goal is to launch a minimum viable product by the end of the semester.

Grades

Because the goal of this class is to teach you how to build software in large teams, PARTICIPATION IS ABSOLUTELY CRITICAL!

Last year when I taught this class for the first time, out of the 45 times that we met that semester, I only had 100% attendance on **one day**. Most days, attendance was closer to 75% or worse. This is **completely unacceptable**. The entire team relies upon all members to show up all of the time. Since everyone relies upon everyone else, if you are not there, you hold **everyone** up.

Therefore, this year there will be a new, and very formal grading policy.

1. If you make it to week 13, you will get an A.
2. If you make it to week 10, you will get a B.
3. If you make it to week 7, you will get a C.
4. If you make it to week 4, you will get a D.
5. If you get "fired" before week 4, you will get an F.

What do I mean by "make it?" Essentially, if the team feels that you are not being professional, and not pulling your share of the load, you will be "fired." That being said, we strive to have a friendly and supportive team, so we'll make every effort, as a team, to make sure that everyone survives until the end of the semester.

> *Working in the professional software world is really fun and exciting! This class is an opportunity for you to take the initiative to make it a positive experience. Imagine what kind of place you would like to work in when you graduate. How can you create your perfect world in this class? I know I'm pretty psyched to see what you guys come up with!*

How do you get fired?

There will be a list of "firable offenses." If you commit any of them, you will be put on probation for a specified period of time. If you make any more mistakes while you are on probation, you will be let go. The firable offenses are as follows:

1. If you have 3 unexcused absences, or if the total of excused and unexcused absences reaches 5, you will be put on probation **for the rest of the semester**. Any more absences after that point will result in your being asked to leave the project.

2. If you harass any of your colleagues, verbally, sexually, or otherwise, you will be put on probation. Your probation will end after you have been through an appropriate training to increase your sensitivity.
3. If you have not satisfactorily completed the independent developer training exercises by the end of week 5, you will be put on probation. If you still have not completed them by the end of week 6, you will be asked to leave.
4. Once we begin sprinting, if you fail to complete your coding tasks in any given sprint, and if the reason is deemed to be due to laziness, or inability to manage your time effectively, you will be put on probation until you have successfully completed the next sprint.
5. You may be put on probation for other types of unprofessional behavior that may arise. If it is not on the list above, you will receive a warning. The second warning will result in probation. The probation will last an amount of time commensurate with the offense.

How NOT to get fired

First of all, you will **NOT** be fired because of a lack of coding skill. If you can't complete a coding task during a sprint because you honestly don't know how to write the code, that's fine–this is a learning experience after all. That being said, if you know that you are in danger of not completing a sprint, and you have not come to anyone for help or assistance, and if you haven't given the team an appropriate amount of notice that you won't hit your target, you may be put on probation. In other words, ALL coders get stuck, pretty much all the time. However, if you are deemed not to be making an effort to learn what you need to learn to get unstuck, the team will not be happy with you.

Here are some positive things you can do to reduce your team's stress level, as well as your own:

1. Spend enough time socializing or otherwise getting to know your teammates that you will feel comfortable coming to them for help if you need it.
2. Look for opportunities to help other people who might be stuck. If you are proactive about helping other people, the chances that they will want to help you in return go up.
3. Get in the habit of keeping notes or a journal about your efforts to get unstuck. Alternatively, write well-documented questions on Stack Overflow. That way we can tell the difference between honestly stuck, and too lazy or disorganized to get your work done on time.
4. Get in the habit of putting in at least an hour a day. Manage your time well, and don't procrastinate.
5. Communicate openly, honestly, and frequently with your team.
6. Show up to all team meetings on time or early, whether during class or outside of class.

Your Goal This Semester

Your goal this semester is to create a functional, supportive team that is fun to be a part of. We are creating a simulated work environment that will attempt to function in as close to a realistic fashion as possible. So look for ways to boost morale, to put people at ease, to make people laugh, and of course, to develop an amazing software application.

Schedule & Flow

The schedule for the course will roughly follow the following timeline:

Week	What the Class is Doing	What You should be Doing
1-5	Orientation and Groundwork	Completing Required Tutorials
6	Pre-Sprint Planning	Planning Poker; Assigning User Stories
7-8	Sprint #1	Development
9-10	Sprint #2	Development
11-12	Sprint #3	Development
13-14	Sprint #4	Development
15	Presentation	Celebration!

Once we reach week 7 we will attempt to have daily standup meetings. On class days these will be held face-to-face, and other days we will use gchat, skype, or some other form of online conferencing system.

Personal Integrity

First:

> **If I catch you cheating, or doing anything else dishonest, you will fail the course. Period.**

Second, that being said, I strongly encourage sharing and collaboration in most every aspect of the course. That means that I think it's a smart idea for you to:

- Download code you find on the web (include the URL of where you found it and some notes about how you got there)
- Download your classmates' code and use it, even before an assignment is due
- Pay someone to help you write code
- Get code from upperclassmen or people in previous semesters
- Ask your neighbor to give you a hint on a question on a test that you're stumped on
- Use whatever notes, websites, books, or other materials you need to complete most any assignment or test

You'll note that many of the above behaviors would be considered "cheating" in many or most other courses. Here are some guidelines I'd like you to follow:

- **Never EVER copy without attribution**
  Even on tests, if someone or something helped you out, acknowledge it. Make notes in your code if you got it from someone or somewhere else. Copying without attribution is stealing and is a breach of integrity. If you got the code off of the web, there should be a URL and some notes about how you found it. If you paid someone to help you write it, say so.
- **Never copy without understanding**
  The point of the class is to learn and understand stuff. Since you don't get any grades on individual tests or assignments, it's pretty stupid to copy something that you don't understand. Think about it. What point could it possibly serve?
- **Be very hesitant to copy an ENTIRE project**
  While there's a lot to be gained by incorporating parts of your classmates' code in your own project, copying someone else's entire project doesn't really provide you much of a learning experience and wastes people's time.
- **Try to figure it out yourself first**
  90% of writing programs is learning how to write them, and this will stay the same throughout your entire programming career. Being a self-sufficent learner is one of the primary goals of the course.
  Code re-use is a HUGE part of hacker culture. What hackers hate more than anything is not understanding stuff. I want you to get a sense for what it's like to be a part of the fun world of professional hackers.

Okay, so what do I consider a breach of integrity worthy of failure?

- Lying about anything to anyone in the class
- It could be as trivial as the reason why you didn't show up for class or do your part of a group assignment. Everybody screws up sometimes. Don't compound the mistake by lying about it. We can forgive mistakes but it's VERY difficult to regain trust once it's broken. Swallow your embarrassment or fear and fess up.
- Stealing anything–this includes copying without attribution

- Stealing is just wrong, and since you have a blanket license to copy most any code you can find, there's no reason not to give people credit for the work they did. Passing someone else's original work off as your own is frankly disgusting.
- Threatening, antagonizing, or intimidating anyone in our learning community
- This is unacceptable behavior and will get you at least fired, if not sued in most every company you'd ever work for.

If you are in doubt about something, please ask your prof. Please feel free to come speak to your prof in confidence about anything in this course that troubles you. So far at JMU I've never had a problem with anyone's integrity (that I know about). Don't be the first group to ruin my perfect record. Thanks!

The Prof

My research mainly involves coming up with pedagogical alternatives that maximize student motivation and learning. Being a tech geek, web-based technology plays a pretty heavy role in what I came up with.

My favorite part of my job is getting to hang out with students and play with technology. Feel free to come see me any time. My info:

Office	ISAT/CS 124
Office Hours	Make an appointment
Mobile	973-495-7736 (calls and texts are ok within reason)
Email	bentonmc@jmu.edu
Facebook	http://www.facebook.com/morgan.benton
Twitter	http://twitter.com/morphatic
Blog	http://www.burningmindproject.org

Important Dates & Deadlines

At the behest of the registrar, a list of dates you may wish to take note of:

- Tuesday, September 6th: Last day of add/drop
- Thursday, September 15th: Last day to add a class with Department Head signature
- Friday, September 16th: Last day to withdraw from JMU with charges canceled

So if I scare you off, get out early. Or conversely, if I turn you on, join us soon!

My academic integrity policy is different from JMU's standard policy, but I will adhere to JMU's standard policies listed on the JMU Syllabus Information for Students page with respect to add/drop, disability accommodations, inclement weather and religious accommodations.