ISAT 345—Software Engineering—Syllabus

Fall 2015

Course Overview

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 Bountify. Bountify aims to help people realize that each of us lives in a world of abundance. The Bountify community believes that all people possess gifts–whether material, physical, intellectual, or spiritual–and that by creating a space where people are able both to share their gifts and articulate their needs, all of us may be

brought closer to living in a society where no one's needs go unmet.

As such, the core functionality of the Bountify website (and its clones) is to

1. Allow people to create profiles in which they
2. Publicize gifts that they have to share, and
3. Articulate needs that are currently unmet through which
4. Matching algorithms can connect people

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

Grades

A great deal of research on learning indicates that really the only good predictor of how much someone will learn is the actual amount of time they spend engaged in learning activities. Therefore, here is how the point accumulation system will work.

1 point ≈ 1 hour of effort (maybe very approximate)

For financial aid purposes, the federal government considers you to be a "full time" college student if you are taking at least 12 credit hours per semester. That typically represents about 4 courses. A full-time job is typically 40 hours per week, so:

40 hours/week ÷ 4 courses = 10 hours/course/week

A semester at JMU is 15 weeks long, so:

10 hrs/course/wk x 15 wks x 1 pt/hr = 150 pts/course

In other words, you need to be spending about 10 hours per week on this course to get the maximum expected value. In order to translate points into letter grades, I'll divide 150 by 5, and assign +/- to the top and bottom 9 points of any grade range so to get into the "A" range, you'll need to plan to spend 8-10 hours per week on this course:

Grade	Points
A	129-150
A-	120-128
B+	111-119
B	99-110
B-	90-98
C+	81-89
C	69-80
C-	60-68
D+	51-59
D	39-50
D-	30-38
F	0-29

How will you earn points?

- **Attend Class and Daily Stand-up Meetings**
  There are 150 minutes of class time each week (2.5 hours) and starting in week 4 we will also be having daily stand-up meetings that last 15 minutes a day (total of 75 minutes per week). I will round up to the nearest hour each week, so during weeks 1-3 you can get 3 points per week for attending class, and after that 4 points per week for attending class and all stand-up meetings.
- **Complete User Stories**
  The bulk of the work you'll be doing this semester is writing code towards the delivery of the project we'll be working on. Code-writing will be broken up into units called "user stories." Each user story will consist of unit tests, and the code that will pass the tests. The number of points you will earn for completing each user story will vary depending on the complexity of the story and will be determined collaboratively by the entire development team. Team members will volunteer for the user stories they want to complete, and you are responsible for volunteering to develop enough to earn the points you need to earn your target grade for the semester.

- **Complete Basic Training**
  In order to be a valued, performing member of our team, you will need to get up to speed on the technologies that we'll be using to implement the project. These tutorials will be collectively worth about 30 points and must be completed by the end of week 3 of the course.
- **Miscellaneous Activities**
  There are likely to be many other opportunities to earn points this semester for completing tasks that are not strictly attendance, tutorials, or development related. These points will be awarded on an ad-hoc basis as the semester progresses.

Here are some other things to keep in mind:

- **Some activities might take longer than the number of points suggests**
  While unfortunate, this is unavoidable. If you feel disgruntled by this, then you will feel the same way that thousands of people do every day who are involved in software projects that have taken longer than estimated. I will not be adjusting the number of points for any of the labs or assignments. Indeed, learning about time estimation is one of the major learning objectives for this course.
- **There is no partial credit**
- **I don't negotiate points**
  I am more than happy to talk about the content, and work with you to overcome challenges you are having. I'm happy to suggest ways in which you might have accomplished a task more efficiently. It is vital for you to understand that the point system is heuristic at best, and completely arbitrary and wildly inaccurate at worst.

As a final note, I know this point system is not perfect. It's kind of like Winston Churchill said about democracy (*Churchill by Himself, p574)*:

> *…it has been said that democracy is the worst form of Government except for all those other forms that have been tried from time to time…*

This point system is the best/worst point system, except for all of the other point systems. 🙂

Important Dates & Deadlines

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

- Tuesday, September 8th: Last day of add/drop
- Thursday, September 17th: Last day to add a class with Department Head signature
- Friday, September 18th: 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.

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!

Schedule & Flow

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

Week	What the Class is Doing	What You should be Doing
1-3	Orientation and Groundwork	Completing Basic Training
4-5	Sprint #1	Development
6-7	Sprint #2	Development
8-9	Sprint #3	Development
10-11	Sprint #4	Development
12-13	Sprint #5	Development
14-15	Sprint #6	Development

Once we reach week 4 we will begin having 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.

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