

SYLLABUS — *DevOps and the modern IT team*

TCMG 412 — Contemporary topics in technology management

Fall 2016

MWF 1:50–2:40 HECC 616

Course Description and Prerequisites

This course will provide a survey of concepts used to manage a modern IT team, with emphasis on the ideas and philosophies encapsulated in the *DevOps* movement. We will discuss the ideas behind Lean and Agile organizations, systems automation and infrastructure as code, delivering modern IT services at scale, and how the IT unit relates to other groups within the organization (business relationship management and IT governance). Prerequisites for this course include the completion of TCMG 272 and 274 with a grade of C or better, and upper-level status within the Technology Management degree program.

Course Objectives

Upon completion of the course, students will have a basic understanding of the concepts and practices behind the DevOps model of IT service delivery. This includes: classifying types of work within an IT unit; systems automation and infrastructure as code; continuous integration/delivery of software; and the importance of communication and culture within an IT group. Additionally, students will have acquired hands-on experience with a variety of tools commonly used in DevOps environments, such as version control systems, virtual machines and containerization systems, infrastructure automation tools, and systems monitoring and reporting tools. The course will provide the opportunity to work in teams to deliver modern IT services, and experience the DevOps model of service delivery from the inside.

Instructor Information

Adam Mikeal
Director of IT, College of Architecture
adam@tamu.edu
(979) 862-3396

Office: Langford A122
Office hours: T 10–12 F 1:30–3

David Sweeney
Director of IT, Division of Student Affairs
sweeney@tamu.edu
(979) 458-9201

Office: John J. Koldus 233X
Office hours: M 4–6 or by appointment

Americans with Disabilities Act (ADA)

The Americans with Disabilities Act (ADA) is a federal anti-discrimination statute that provides comprehensive civil rights protection for persons with disabilities. Among other things, this legislation requires that all students with disabilities be guaranteed a learning environment that provides for reasonable accommodation of their disabilities. If you believe you have a disability requiring an accommodation, please contact Disability Services, currently located in the Disability Services building at the Student Services at White Creek complex on west campus or call 979-845-1637. For additional information, visit <http://disability.tamu.edu/>.

Academic Integrity

We expect all students to abide by the Aggie Honor Code: *An Aggie does not lie, cheat, or steal, or tolerate those who do.* For additional information, visit <http://aggiehonor.tamu.edu/>.

Grading Policies

Final grades will be allocated on a standard 100-point scale (A=90-100, B=80-89, C=70-79, D=60-69, F<60). Attendance and participation in class is a critical part of your grade. The University views class attendance as the responsibility of an individual student. University rules related to excused and unexcused absences are located online at <http://student-rules.tamu.edu/rule07>. The total grade is comprised of four main components:

Attendance and class engagement	40%
Weekly ERFAs	20%
Lab projects / exercises	30%
Leadership and character	10%

Attendance and class engagement – 40%

A primary skill needed to work in a DevOps environment is the ability to work within a group, and to be considered a dependable and productive member of a team. This means you must be prepared, be on time, and be engaged. Accordingly, a substantial part of your grade is based on timely attendance and participation in class and on your team. This grade component will be equally divided into four parts:

1. **Attendance.** This is simple: show up to class (unless you're sick – then stay home. Nobody wants to catch that... =P).
2. **Assigned reading.** There will be brief quizzes over each assigned reading on the due dates listed in the course schedule.
3. **In-class and online participation.** The majority of the classes will be conducted seminar-style, with a discussion over the assigned reading. Engagement in these discussions is important for a productive and interesting semester.
4. **Peer evaluations.** After each group project all team members will complete an evaluation for each one of their teammates. The average will be used to determine this grade.

Weekly ERFAs – 20%

Each week for the first fourteen weeks of the semester, you will have an online self-assessment called an *ERFA*. ERFAs encourage you to *Engage, Reflect, Find*, and *Apply* what you are learning in this course. The weekly ERFA assessments open in eCampus at 5pm on Friday afternoon and close at 11:59pm on Sunday night, and ask questions about that week's readings, lectures and activities. The questions are identical each week. Part of the assessment deals with us (the instructors) asking you for feedback on how things are going – you won't be graded down if you are thoughtfully critical of our lectures or teaching. You **will** be graded down if the quality of your self-reflection is poor. Otherwise, this should be an easy part of your grade.

Lab projects and exercises – 30%

Hands-on exercises are assigned nearly every week of this course. Some of these are individual assignments that involve learning tools and applications common to the DevOps workflow. You should use the free period on Wednesday to work on these assignments. Some involve video instruction from Lynda.com, while others reference tutorials available from external sites. You are expected to learn the material outside of the normal class time, but be prepared to demonstrate your progress during class on Friday. There are four group projects throughout the semester. You will be assigned to a group for these projects, and graded collectively.

Leadership and character – 10%

The Texas A&M University purpose statement says: *to develop leaders of character dedicated to serving the greater good*. One of the things you'll notice about this curriculum is the amount of time spent on "soft skills", like

team-building, communication, and empathy. That's because DevOps is as much a cultural movement as it is a technological one. In recognition of this fact, the final 10% of your grade is determined by the leadership qualities that you demonstrate in class and within your project teams throughout the semester. The criteria for this determination are drawn from the [core values](#) of Texas A&M:

- **Selfless service.** Do you help a struggling teammate instead of complaining about them? Do you coach an unproductive or marginalized teammate back into the group, or just write them off?
- **Integrity.** Can your teammates count on you to follow through? Are you consistently prepared and on time?
- **Respect.** Do you actively work to ensure your classmates feel welcome and respected? Do you help to create an environment of trust and courtesy in the classroom?
- **Excellence.** Do you consistently strive to achieve your best work, and encourage others in your team to do the same?

NOTE: There is no late work acceptance policy — assignments will be judged by their state at the time of their due date. Learning to work within a team and plan ahead for contingencies is a critical skill in the workplace, especially so for someone working in a DevOps environment.

Texts and Resource Materials

REQUIRED

Effective DevOps by Jennifer Davis and Katherine Daniels

ISBN: 1491926309 Available online: <http://proquest.safaribooksonline.com/9781491926291>

The Phoenix Project: A Novel about IT, DevOps, and Helping Your Business Win by G. Kim, K. Behr, and G. Spafford

ISBN: 0988262509 Available online: <http://proquest.safaribooksonline.com/9781457191350>

The Five Dysfunctions of a Team by Patrick Lencioni

ISBN: 0787960756 Available online: <http://proquest.safaribooksonline.com/9780787960759>

The Art of Business Value by Mark Schwartz

ISBN: 1942788045 Available for Kindle: <https://www.amazon.com/dp/B01DOGOBJ0>

Beyond Blame by David Zwieback

ISBN: 1491906413 Available for Kindle: <https://www.amazon.com/dp/1491906413>

The Human Side of Postmortems by Dave Zwieback

ASIN: B00CLH38CM Available for Kindle: <https://www.amazon.com/dp/B00CLH38CM>

RECOMMENDED

Infrastructure as Code by Keif Morris

ISBN: 1491924357 Available online: <http://proquest.safaribooksonline.com/9781491924334>

Lean Enterprise: How high performance organizations innovate at scale by J. Humble, J. Molesky & B. O'Reilly

ISBN: 1449368425 Available online: <http://proquest.safaribooksonline.com/9781491946527>

Kanban: Successful Evolutionary Change for Your Technology Business by David J. Anderson
 ISBN: 0984521402 Available for Kindle: <https://www.amazon.com/dp/B0057H2M70>

Continuous Delivery and DevOps – A Quickstart Guide - Second Edition by Paul Swartout
 ISBN: 1784399310 Available online: <http://proquest.safaribooksonline.com/9781784399313>

Site Reliability Engineering: How Google Runs Production Systems by Beyer, Jones, Petoff, and Murphy
 ISBN: 149192912X Available online: <http://proquest.safaribooksonline.com/9781491929117>

Antifragile Systems and Teams by Dave Zwieback
 ASIN: B00KQVXTLO Available online: <http://proquest.safaribooksonline.com/9781491908587>

Leading the Transformation by Gary Gruver and Tommy Mouser
 ISBN: 1942788010 Available online: <http://proquest.safaribooksonline.com/9781457191374>

Course Schedule

DATE	TOPIC	ASSIGNMENTS DUE
WEEK 1 – Course overview and setting the stage		
Monday, 29 Aug	Syllabus; class expectations	
Friday, 2 Sept	The modern IT team; what is IT service delivery?	Reading: <i>The Phoenix Project</i> , ch. 1-8
WEEK 2 – When IT fails		
Monday, 5 Sept	Theory of constraints; wait/busy times	Reading: <i>The Phoenix Project</i> , ch. 9-16
Friday, 9 Sept	The four types of work; guarding constraints	Exercise: Group project #1
WEEK 3 – Rethinking IT service delivery		
Monday, 12 Sept	The three ways; total systems thinking	Reading: <i>The Phoenix Project</i> , ch. 17-26
Friday, 16 Sept	The improvement kata	Reading: <i>The Phoenix Project</i> , ch. 27-35
Self-Study (use Wednesday)	VirtualBox, Vagrant, and Linux Video courses in TCMG 412 playlist on Lynda.com – http://goo.gl/aRQ190	

WEEK 4 – What is DevOps?		
Monday, 19 Sept	Introducing terms; DevOps as folk model; the history of DevOps; <i>Introducing Slack</i>	Reading: <i>Effective DevOps</i> , ch. 1-3
Friday, 23 Sept	Foundational concepts; misconceptions and anti-patterns; the 4 pillars of DevOps	Reading: <i>Effective DevOps</i> , ch. 4-6
Self-Study (use Wednesday)	Version control systems, git, and GitHub.com Lynda.com – http://goo.gl/aRQ190	
WEEK 5 – Collaboration: Individuals working together		
Monday, 26 Sept	Pillar 1: collaboration; individuals and the group; conflict resolution; trust and empathy	Reading: <i>Effective DevOps</i> , ch. 7
Friday, 30 Sept	Collaboration misunderstandings and troubleshooting	Reading: <i>Effective DevOps</i> , ch. 8 Exercise: Group project #2
WEEK 6 – Affinity: from individuals to teams		
Monday, 3 Oct	Pillar 2: affinity; team culture/cohesion; diversity; hiring; org structure	Reading: <i>Effective DevOps</i> , ch. 9
Wednesday, 5 Oct	<i>Guest lecture:</i> Benjamin Liles, Site Reliability Engineer at Google	
Friday, 7 Oct	Affinity misconceptions and troubleshooting	Reading: <i>Effective DevOps</i> , ch. 10
Self-Study (use Wednesday)	Programming basics: the Ruby language Lynda.com – http://goo.gl/aRQ190	
WEEK 7 – Team building		
Monday, 10 Oct	Working with a dysfunctional team; the importance of trust; stages of team development; teams and sports	Reading: <i>The Five Dysfunctions of a Team</i> , ch. 1-26
Friday, 14 Oct	Conflict-avoidance; commitment; accountability; inattention to results	Reading: <i>The Five Dysfunctions of a Team</i> , ch. 27-44 Exercise: Group project #3
WEEK 8 – Tools: accelerators of culture		
Monday, 17 Oct	Pillar 3: tools (<i>Automate all the things!</i>); development; automation; monitoring; how tools affect culture	Reading: <i>Effective DevOps</i> , ch. 11-12
Friday, 21 Oct	Tools misconceptions and troubleshooting	Reading: <i>Effective DevOps</i> , ch. 13
Self-Study	Containerization and Docker	

(use Wednesday)	Docker Basics on Lynda.com – http://goo.gl/aRQ190 Fundamentals and Operations from https://training.docker.com/self-paced-training	
WEEK 9 – Scaling and preparing for growth		
Monday, 24 Oct	Optional Road Trip: attend Amazon Dev Days (AWS mini-conference): https://aws.amazon.com/events/devday-austin/ Contact <i>David</i> for details	
Wednesday, 26 Oct	Understanding scaling; effect on technology; effect on teams and organizations	Reading: <i>Effective DevOps</i> , ch. 14
Friday, 28 Oct	Scaling misconceptions and troubleshooting	Reading: <i>Effective DevOps</i> , ch. 15
Self-Study	Learning automation with <i>Chef</i> Online tutorials from https://learn.chef.io/tutorials/ <i>Assigned:</i> New to Chef; Learn the basics; Manage a node; Develop locally	
WEEK 10 – The Lean Enterprise		
Monday, 31 Oct	The Improvement Kata: continuous process innovation	Reading: <i>Lean Enterprise</i> , ch. 6
Friday, 4 Nov	Kanban and agile; lean manufacturing	Reading: Selections from <i>Kanban (Anderson)</i>
Self-Study (use Wednesday)	Learning automation with <i>Chef</i> Online tutorials from https://learn.chef.io/tutorials/ – <i>Manage a basic web application</i>	
WEEK 11 – DevOps is Culture		
Monday, 7 Nov	Understanding culture: values, prohibitions, myths, rituals, shared knowledge	Reading: <i>Effective DevOps</i> , ch. 16-17
Friday, 11 Nov	Putting everything together; systems health	Reading: <i>Effective DevOps</i> , ch. 18 Exercise: Group project #4
WEEK 12 – Blame free culture		
Monday, 14 Nov	Postmortems and just culture; Cynefin framework; prioritizing team growth over feelings	Reading: <i>Beyond Blame</i> , all chapters
Wednesday, 16 Nov	Psychological biases; conducting a postmortem	Reading: <i>The Human side of Postmortems</i>
Friday, 18 Nov	<i>Guest lecture:</i> Gene Kim, author of <i>The Phoenix Project</i>	
Self-Study (use Wednesday)	Learning automation with <i>Chef</i> Online tutorials from https://learn.chef.io/tutorials/ – <i>Testing code</i> and <i>Manage your own Chef</i>	

WEEK 13 – On leadership		
Monday, 21 Nov	Guest lecture: Col. Byron Stebbins, Chief of Staff for Texas A&M Corp of Cadets	Assignment: Final project proposal
Friday, 25 Nov	THANKSGIVING HOLIDAY – <u>NO CLASS</u>	
WEEK 14 – It's about the business, stupid		
Monday, 28 Nov	What is business value? Who decides?	Reading: <i>The Art of Business Value</i>
Friday, 2 Dec	Guest lecture: Philip Mattingly, Senior DevOps Engineer at Endgame	
Self-Study (use Wednesday)	Learning automation with <i>Chef</i> Online tutorials from https://learn.chef.io/tutorials/ Continuous delivery and build automation: <i>Get started with Chef Automate</i>	
WEEK 15 – Wrapping up		
Monday, 5 Dec	<i>In-class presentations:</i> final group projects	
Wednesday, 7 Dec		
FINAL EXAM – Tuesday, Dec 13 @ 3:30–5:30pm		

GROUP PROJECT #1: Deploy a webapp, oldstyle

The class will be divided into groups of 8-10 students each. These groups will be split into Dev and Ops teams. Each group is responsible to deliver a running Ruby on Rails webapp (<https://github.com/locomotivecms/engine>) on Friday. The application must be modified so that the appearance of the interface is visibly different from the default code as published on GitHub.com. Additionally, since this is a CMS, there should be at least three pages with original content in the final product.

RULES:

1. The Dev team is responsible for any coding. They will make the necessary alterations to the interface design, and also create the content that will be duplicated in the production environment.
2. The Ops team is responsible to create and manage the production environment that will host the Rails app (including OS and any databases, etc). This can be done on a VM from the student virtual environment, or using another server to which you have access.
3. The Dev team cannot touch the production boxes.
4. The Ops team must take the prepared application and related data from the Dev team and deploy it, and the result must match what the Dev team sees on their development environment (both the production and development environments will be examined to judge success).
5. Both teams will be assigned a leader from within the team. Cross-team communication may only happen via email, and only through each team leader (no direct conversations between Dev and Ops team members!)

GROUP PROJECT #2: GitHub and collaboration with code

For this project, you will use the same four groups from Project #1, with the change that we will no longer be split into Dev or Ops teams (just Team A, Team B, Team C, and Team D). The goal of this project is to practice the workflow of social coding – learning to collaborate within a group on a common set of data. Each team will deliver a simple website that is hosted on Github.com using the Github Pages service. Github Pages is typically used to host an informational website about a technical project; in this case, there is no technical project behind the website, but you can just write a placeholder file (like a README file) to have something in the master branch.

STEPS

1. Create a new repository for your project and add some placeholder data (like a README file, or an code license). You might want to consider creating a Github organization to help you manage the people in your team and privileges around your repository.
2. Use the **Automatic Page Generator** to create a sample website for your project: <https://help.github.com/articles/creating-pages-with-the-automatic-generator/>.
3. Start creating content. It doesn't matter what you put here, but it should be somewhat substantive, and look polished. There should be a minimum of three pages on the site, and some images. You can create a site for our course, or a pretend site for the (non-existent) technical project you created earlier, or even something completely unrelated to our class topic (like a site about a band or TV show, or collaboratively-written fiction). It's really up to your group.
4. You can assign various group members to different sections or pages on the site, but an important part of social coding is learning to resolve conflicts that arise when multiple people make conflicting changes to the same file. I will be looking at the Github logs to see how these situations are resolved. You should use the collaboration tools provided by Github to assist (such as Pull requests, and the **Conversation** mechanism that is provided. If these situations don't naturally occur in your project, **engineer them**.
5. We will be looking at the sites as a class next Friday. I will be grading this project on two criteria: how much you individually contributed to the project (as evidenced by the Github records), and how broadly your team used the version control features of git and the collaboration features available in Github.

GROUP PROJECT #3: Writing an app in Ruby

GROUP PROJECT #4: Deploy a webapp, DevOps style