# 17-313: Foundations of Software Engineering

Fall 2021 Midterm Exam

Michael Hilton, and Rohan Padhye

Name: ______________________________

Andrew ID: ______________________________

**Instructions:**

- Not including this cover sheet, your exam should have 12 pages in total: 11 pages of questions, and 1 page of appendix. Make sure you're not missing any pages. Write your full name and **Andrew ID** on at least this page, if not all others.
- Most questions in this exam refer to a scenario, described in the appendix. We encourage you to read it first. You may detach the appendix; the appendix will not be considered in grading, thus keep your answers on the question pages.
- Write concise, careful answers. Short and specific is much better than long, vague, and rambling, and grading will reflect this. You have enough time to write clear and readable responses. Bullet points and phrases are acceptable. Spend time to consider how best to present your answers, including citing examples to make your points more concretely.
- Clearly indicate and write your answers in the space provided below each problem. We cannot give you points for answers we cannot find or read.
- The exam has 5 multi-part questions with a maximum score of 80 points. The point value of each problem is indicated. We planned the exam to allocate approximately one minute per point.
- If you do not know the answer to a question, you may leave it blank and receive the indicated number of "No-nonsense" points (abbreviated *NN* for the rest of this exam). This may be an improvement over the zero (0) points that may be awarded an incorrect answer. The number of available *NN* points varies per question. If you start to answer a question and then change your mind, you may invoke this rule by crossing out your answer and prominently writing "LEAVING THIS BLANK." We will not read partial answers to questions on which you invoke this rule.
- You may consult one sheet of paper (8.5x11in, double sided, typed or handwritten) with notes. You may not use books, a calculator, cell phone, laptop, or any other electronic or wireless device.
- Good luck!

## Question 1: Process and Planning (14 points)

**Appendix A describes a scenario that will be referenced throughout the exam. You should familiarize yourself with the scenario before you continue**

(a) (4 points) *(1 NN)* Cups & Pups (Appendix A) decides that it is time that they start accepting cryptocurrency, specifically all dog related cryptos: *Dogecoin, Shiba Inu, Dogelon Mars, DogeCash, and Baby Dogecoin.* You will implement this via API calls to a major cryptocurrency platform. However, the current system only accepts major credit cards. There are a lot of hard coded values, and so you and your team decide to completely reimplement the payment subsystem. To ensure you are making progress, you ask the team to define intermediate milestones. Describe one good intermediate milestone for the subsystem.

(b) (4 points) *(1 NN)* The Boeing case study involved a discussion of how process and culture concerns led to software failures. Briefly describe one of those issues.

Question 1 continues. . .

(c) (6 points) *(2 NN)* What is one process intervention you could implement in your company to avoid the same kind of failure as in the previous question?

*Writing below this line is permitted but discouraged.*

Exam continues. . .

## Question 2: Requirements (13 points)

In the scenario, one feature you're considering implementing is forcing users to log in via a facebook account. This will simplify the user management, and make the authentication more secure, but will also allow the marketing department learn more about your customers.

(a) (5 points) *(1 NN)* You quickly realize that you have several potential conflicts between the goals of your Public Relations department (who are concerned that some customers don't have a Facebook Account), your salesperson (who wants to provide as much user information to advertisers as possible), and your developers (who believe that current authentication mechanisms are not secure and leave the app vulnerable).

Describe one of these conflicts in terms of the conflicting goals (or subgoals) involved, and give one way to resolve it:

(b) (8 points) *(2 NN, 1 per story)* One way to evaluate the quality of user stories is according to the INVEST principles (Independent, Negotiable, Valuable, Estimatable, Small, Testable). For 'independent' and 'testable' each, write a user story that is generally sensible in the Cups& Pups scenario, but violates (only) that principle. Afterward briefly explain why this user story violates the principle and provide an improved version of the same user story. Use the common *"As a [role], I want [function], so that [value]"* template.

i. User story violating (only) the *'independent'* principle:

Briefly explain why it violates the principle:

Fixed version of the same user story:

Question 2 continues. . .

ii. User story violating (only) the *'testable'* principle:

Briefly explain why it violates the principle:

Fixed version of the same user story:

---

*Writing below this line is permitted but discouraged.*

Exam continues. . .

## Question 3: Metrics and Measurement (16 points)

The old code base is a real mess! Your team begins to think about how to improve it, what sorts of process to establish to improve its quality over time.

(a) (5 points) *(1 NN)* EasyOrder has a license for a tool that indexes the entire repository, observes the CI, and has a plugin for your IDE. The tool creates a relative ranking to judge the number of lines of code written by each developer. This allows you to use the quantity of code written by each developer over a given time as a usable metric. Your CTO suggests you use this tool to identify developers who are writing the least amount of code, and then provide them mentoring and extra training. Do you think this is a good idea? Why/why not?

(b) (6 points) *(1 NN per part)* For each of the following quality attributes, give a good metric that you can use to determine if the implemented cryptocurrency payment module is satisfying the associated quality requirement:

i. *Performance:*

ii. *Overall Usability:*

Question 3 continues…

(c) (5 points) *(1 NN)* Your teammate gets cranky during this discussion, and argues that “overall usability” cannot be verified or measured using metrics and must instead be left to the intuition of the designer. Your teammate says “none of these options really capture the core idea of ‘usability’, it’s pointless!” Do you agree or disagree? Why or why not?

*Writing below this line is permitted but discouraged.*

NOTE: Since Software Engineering for ML has not been covered yet this semester, please ignore this question.

## Question 4: Software Engineering for Machine Learning (12 points)

Your CEO decides that your company is going to roll out a feature to identify if customers are wearing their masks correctly or not. Your app will use the camera to take pictures, and then use machine learning to determine if they are wearing a mask correctly. To accomplish this, you will work closely with a new Machine Learning expert that your company has hired. They will be responsible for choosing and tuning the model. You are tasked with integrating the ML component into the rest of the application.

(a) (8 points) *(2 NN, must leave both parts blank)* Before you begin to develop this mask detection feature, you decide to evaluate its ethical implications.

i. Do you have any ethical concerns with how this new feature may be used? Why or why not?

ii. What mitigations do you propose to help guard against ethical concerns?

(b) (4 points) A pretrained model can be purchased from a vendor who says it achieved 80% accuracy when tested in a local ride sharing app.

*(1 NN)* Is this good enough? Why or why not?

Exam continues. . .

## Question 5: Architecture (25 points)

The great thing about your app is the feature customization that you provide to customers. However, behind the scenes, your app is just a monolith that includes all features that you have developed for every customer. Your app selectively displays features based on the restaurant ID in URLs. At the backend, it's just one monolithic web application replicated across a handful of servers. This architecture has served you well for the handful of customers you have in Pittsburgh. However, your company now plans to expand your operations to other cities and you anticipate more customers and their corresponding feature requests. You consider the following options: (There may be other options; stick with these for the purposes of this question.)

- **Option 1:** Stick with the monolithic architecture, but refactor the build system so that you can compile customized versions of the web app that select only the required features per customer. Then, deploy these customized web apps in separate Docker containers on the cloud.
- **Option 2:** Transition to a microservice based architecture, with a special 'features' service for retrieving the features for a given customer, and several feature-specific microservices. Deploy each microservice in the cloud with elastic (i.e., on-demand) scaling, so that features that are used by multiple customers get replicated more often as demand rises.

Question 5 continues. . .

(a) (9 points) *(3 NN, 1 per quality attribute)* Architectural reasoning often involves tradeoffs, often with respect to quality attributes or non-functional properties. Identify three important quality attributes that might be affected, and describe (briefly) how each option affects it.

| Quality | Effect of Option 1 | Effect of Option 2 |
|---|---|---|
| | | |
| | | |
| | | |

Question 5 continues. . .

(b) Let's assume that you have taken the decision to transition to microservices.

i. (6 points) *(1 NN, must leave all parts blank)* Your existing application was developed monolithically. How would you transition to microservices? Do you recommend implementing new features as microservices and slowly transitioning, or re-writing the existing application from scratch right now? Give one argument in favor of each position and make a recommendation.

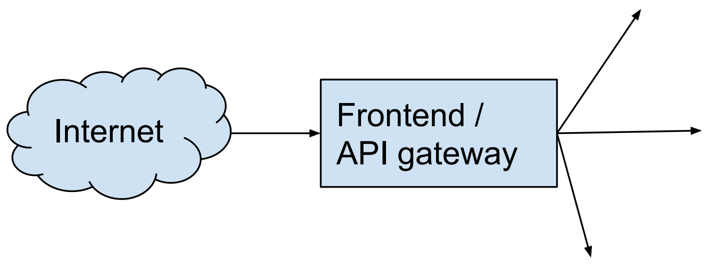
Argument for incremental transition:

Argument for full rewrite:

Recommendation (with brief justification in the context of the scenario):

ii. (6 points) *(1 NN)* Draw an architecture diagram (by completing the template below) for a microservice-based instantiation of your web app with the Cups & Pups customer in mind. Include all features discussed in the previous questions (Q2-Q5). We don't require a specific format for the diagram, but it should be clear and understandable.

Question 5 continues…



iii. (4 points) *(1 NN, must leave both parts blank)*: Design documents consider goals, but also non-goals. As a part of writing a design doc for potentially moving to microservices, give one goal, and one non-goal you would include in your design doc.

goal:

non-goal:

End of questions.

## A Appendix: Scenario description

You work for EasyOrder, a small, fictional technology company specializing in supporting QR code ordering web apps (scan-to-order) for local restaurants. With these apps, restaurant customers can simply scan a QR code with their phone camera to open the menu and pay the bill, all without touching a paper menu or interacting with a server. In addition, it allows restaurants to build a database of their customers' order histories and let businesses integrate more tools for tracking, targeting, and analytics.

There are many QR code ordering systems available in the market. To differentiate EasyOrder's system from those of big technology companies, EasyOrder offers custom feature addition to the ordering system, with an additional contract. As restaurants with varying customer sizes and items have different needs, many restaurants make the additional contracts to add new features specific to their pipelines.

One of EasyOrder's existing customers, Cups & Pups, has recently reached out to add new features. Cups & Pups is a local dog cafe with a private patio, where customers can bring their dogs to spend time together. Cups & Pups serves brunch menus and beverages like a regular cafe, but it also serves dog treats so that both humans and dogs enjoy their time together. (However, due to COVID, human food service is suspended due to the indoor mask requirements, and they only serve dog treats now.)

The QR code ordering system at Cups & Pups currently works as follows (this is fairly standard): Customers come into Cups & Pups and take a seat. They manually scan the QR code on each table using the app. As each QR code is unique to each table, the app can automatically identify the table, without asking the customers to specify that. The app displays the menu, and the customers can place their orders and pay using the app. Once the payment is confirmed, the kitchen gets the order information, and once it is ready, the staff takes the order to the table. Before the customers leave Cups & Pups, they can tip within the app, and/or give feedback via a text form.

End of exam.