17-313: Foundations of Software Engineering

Fall 2020 Midterm Exam

Claire Le Goues, Michael Hilton, and Christopher Meiklejohn

Name: ______________________________

Andrew ID: ______________________________

Instructions:

- Not including these cover sheets, your exam should have 12 pages in total: 10 pages of questions, and 2 pages of appendix.
- Please don't write your name or Andrew ID on any of the answer pages.
- The exam is available for a 24 hour period corresponding to Thursday, October 22, Pittsburgh time. Please complete and submit it by 11:59pm EDT, October 22nd, 2020 to Gradescope.
- You may complete the exam in any of several ways:
  - Print it out, complete it by hand (write neatly!) and then either scan the pages or take (CLEAR) photographs of it to upload to Gradescope.
  - Type your answers on the PDF itself using a PDF editor.
  - Type your answers in some other text editor, like Word or a latex editor. To make it easier to indicate answer locations on Gradescope, you should probably put one question per page.
  - Handwrite all of your answers, neatly (again, one question per page is probably easiest).
  - A mix of the above — some questions may benefit from diagrams that can be hand-written and then scanned/uploaded, while others may be typed.
- When you submit it to Gradescope, you *must* indicate which pages correspond to which question in Gradescope (and doing so correctly is worth 5 points!). This is very important! We can't give you points for answers we can't find. Make sure you leave yourself time to get the exam entered into Gradescope properly.
- Most questions in this exam refer to a scenario, described in the appendix. We encourage you to read it first and refer to it separately. The appendix isn't considered in grading and doesn't need to be uploaded to Gradescope.
- This exam is designed to be completed in 80–90 minutes total. We encourage you to track when you start the exam and try not to take much longer than that; it shouldn't ruin your whole day.
- The amount of space allocated per question (as well as the number of points) suggests approximately the amount of space we think you need to answer the question well, assuming you're writing relatively neatly by hand. Bear this in mind if you choose to type up your answers and try not to go overboard. Aim for concise and careful answers; we much prefer short/specific to vague and rambling. Bullet points/clear phrases are acceptable.
- The exam has 6 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.

  This No-nonsense points system isn’t designed to be gamed—don’t overthink it. If you have no idea, just leave it blank.

- You may reference any notes or course materials you’d like. You may not work with or discuss your answers with others in the course. We are not proctoring the exam; we are choosing to trust you. We will note, however, that collusion is pretty easy to spot on an exam like this.

- The course staff will be available on zoom during the regular course time for any clarifying questions you may have (like we’d be if we were taking this test in the classroom). We will also be available on Slack throughout the day as normal and will try to be especially prompt in responding to your questions. Please DM (all three of!) us rather than asking questions in the publicly-visible channels.

- Contact us if you have any concerns not addressed above.

- Good luck!

Question	Points	Score
Followed instructions	5	
Process and Planning	14	
Requirements	13	
Metrics and Measurement	15	
Software Engineering for Machine Learning	14	
Architecture	19	
Total:	80	

Question 1: Followed instructions (5 points)

No NN points given for this question!

Did you upload to gradescope correctly, indicating where your answers for each question can be found?

Exam continues…

Question 2: 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)* One major problem with the current system described in the scenario (Appendix A) is that the system for checking out produce (e.g., vegetables) sold by weight is effectively unusable. The subsystem accesses the scale hardware to read the weight, and provides a user interface to either enter an item's code, or look it up using pictures or text search. It also integrates with a backend database containing products with numbers, pictures, and prices. You decide to completely reimplement this 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 2 continues…

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

Writing below this line is permitted but discouraged.

Exam continues. . .

Question 3: Requirements (13 points)

In the scenario, one feature you're considering adding to the system is advertising and personalized coupons. Based on a user's purchase history, the kiosk will both (1) display ads while the user scans, for products it predicts the user might like, and (2) print targeted coupons predicted to be useful for the buyer in the future to bring them back to the store.

(a) (5 points) *(1 NN)* You quickly realize that you have several potential conflicts between the goals of your Public Relations department (who observe that some of the items sold in the pharmacy might be sensitive), your saleswoman (who wants to provide as much user information to advertisers as possible), and possibly even your backend engineers (who have planned a very simple database schema mapping user IDs to information, to maximize maintainability and extensibility).

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 'estimable' and 'small', each write a user story that is generally sensible in the Kiosk 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 *'estimable'* principle:

Briefly explain why it violates the principle:

Fixed version of the same user story:

Question 3 continues…

ii. User story violating (only) the *'small'* 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 4: Metrics and Measurement (15 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) (4 points) *(1 NN)* Giant Eagle has a license for a tool that indexes the entire repository and links it with CI results and bug reports to create profiles for each developer. The tool creates an index from 0–100 to judge the code quality of the code produced by each developer, based on how many bugs and style violations were detected by tools like FindBugs, and how many issues were reported in the bug tracker. Your data scientist suggests you use this tool to identify developers with the buggiest code, who could then be mentored or provided 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 self-checkout system is satisfying the associated quality requirement:

i. *Kiosk performance:*

ii. *Overall Usability:*

Question 4 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.

Exam continues…

Question 5: Software Engineering for Machine Learning (14 points)

The targeted coupons/advertising feature described previously is identified as a possible place to apply a machine learning-based solution that aggregates buying data across many users to choose which advertisements to show or coupons to print. To accomplish this, you will work closely with a new Machine Learning expert that your company has hired. They will be responsible for your choosing and tuning the model. You are tasked with collecting the data that the model will use, ideally focusing on data that is easy to collect.

(a) (8 points) *(2 NN, must leave both parts blank)* Before you begin to develop your 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) You are responsible for adding the ML functionality into the existing system.

i. (3 points) *(1 NN)* Where should the model live?

ii. (3 points) *(1 NN)* How often should it be updated?

Exam continues. . .

Question 6: Architecture (19 points)

(a) *(1 NN per sub-part)* Appendix B shows an architecture diagram for the current overall system that you are improving/replacing.

Is the diagram suitable for reasoning about the software system's:

i. (3 points) *Security*? Why/why not?

ii. (3 points) *Portability* (adaptability to hardware changes)? Why/why not?

(b) (5 points) *(1 NN)* Your telemetry indicates that the store server's *availability* is not sufficient for your system's needs. What is one way you could improve the availability of the system? You can describe this in text/prose, but a labelled diagram is also acceptable if you prefer.

End of questions.

(c) (8 points) *(1 NN per quality)* The software at present is optimized to run on individual store servers, as shown in Appendix B, with some interaction with external services/clusters. You host the server-side computations on hardware in-house, for everything but payment; the current application is modular, but monolithic.

Your team is exploring alternative, non-monolithic architectures for the system. For example, consider the following alternatives:

- Option 1: One of your developers has a robotics background, and observes the effectiveness of an event-based architecture (independent processes communicating through an event bus such that processes produce events without needing to know who is consuming them and can consume them without knowing who produces them) for cyber-physical systems (which the self-checkout system is).
- Option 2: Another of your teammates is very fashionable, and argues that you transition the system to be purely microservice-based.

Architectural reasoning often involves *tradeoffs*, often with respect to quality attributes or non-functional properties. Identify two 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 6 continues. . .

A Appendix: Scenario description

Like many grocery store chains, Giant Eagle has invested heavily in self-checkout counters. However, they are very frustrated with their current system (see below). Previous attempts to improve the situation with the system vendor have been unsuccessful. Giant Eagle owns the system (hardware and software), and has decided to not renew the maintenance contract with the previous provider (a consultant/contractor) and instead bring development in-house, to your team, to improve (and possibly rewrite) the current system and develop new features.

This decision was the result of considerable debate amongst the higher-ups at Giant Eagle, and so the pressure on your team to perform is high. You commit to management to produce a beta release of an updated version of the software, with some (unspecified!) cool new features, to be deployed to two stores as part of a pilot program, within 6 months. Your team is part of the overall development organization within Giant Eagle which overall has about 200 developers, including experienced senior developers and many recent hires of diverse backgrounds. Your new 8-person team, of which you are appointed the head, is formed for this project, and is given extensive autonomy on process and design/implementation as long as they deliver results.

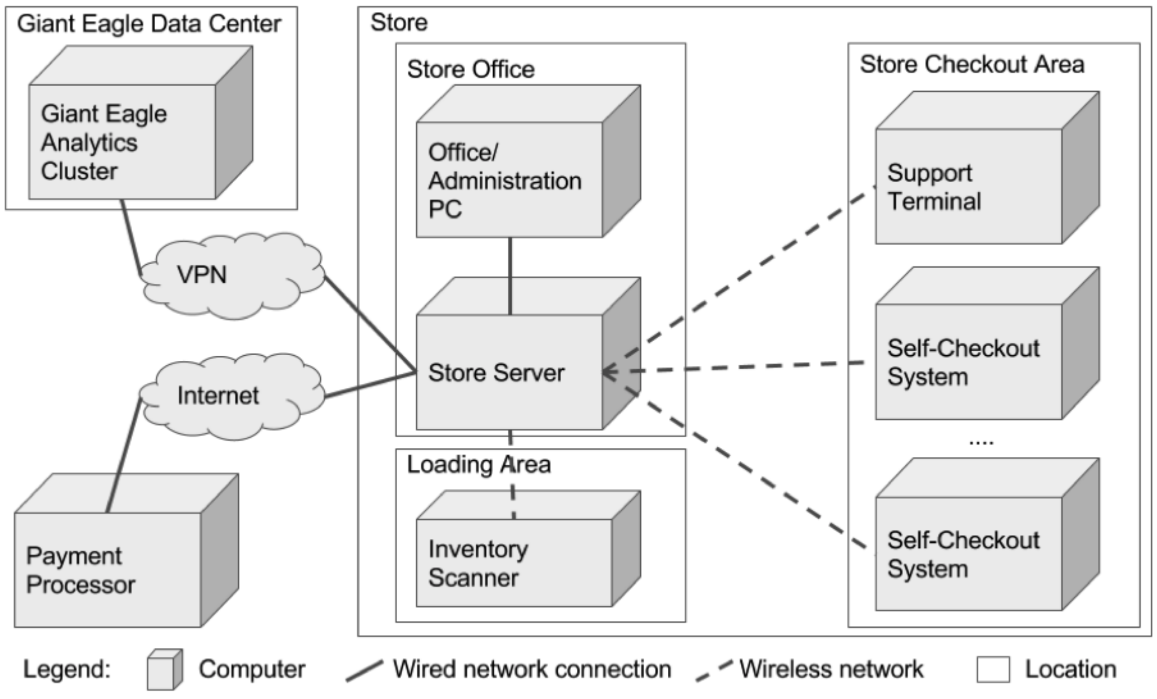
The self-checkout systems currently work as follows (this is fairly standard): *Customers take their purchases to a kiosk at which they manually scan the barcodes on each item. Once an item is scanned, it is placed in a bag on a scale next to the scanner. The system confirms that the weight of the item matches what was scanned, and further that no items are placed in the bagging area without being scanned. If customers do not want to bag an item (and place it directly in the cart instead), they must indicate as such immediately after it is scanned For items without barcodes (produce, such as apples, or certain bulk items), customers either manually enter a PLU code, found on a sticker on the item, or search a listing to look it up. The item is then placed on a second integrated scale, and the price is computed from the weight.*

Giant Eagle initially expected that self-checkouts kiosks would save money by reducing the amount of checkout staff (a single staff member can supervise 6–8 kiosks). However, this is not bearing out in practice. First, customers are very slow as compared to experienced cashiers, both due to inexperience and to the fact that the current system lags, and has a very unintuitive UI. Customers are also frustrated by the numerous exceptions that require staff intervention. For example, if a customer accidentally places her purse on the bagging scale, the system detects an unexpected item, and locks up. Finally, self-checkout lines are a major source of shoplifting, because their fraud detection is unsophisticated. For example, the system cannot tell the difference between organic and conventional apples. A confused customer can accidentally select the wrong version; a malicious customer can do so deliberately. Either way, the customer underpays.

In addition to providing the usual (but improved) functionality of the existing system, the new self-checkout software should be extensible, because Giant Eagle envisions several new features to be added incrementally after the new software is released. These include cameras to identify items without a bar code, smartphone integration to replace or enhance loyalty cards and credit card payment systems, integration with a Giant Eagle shopping app showing past transactions and personalized special offers, and much more.

Question 6 continues. . .

B Architecture Diagram from Existing System



Textual description: Each store has its own infrastructure with a central server in a locked office. Self-checkout systems are placed in the checkout area and are connected to the store's server. Payments are delegated through the server to a payment processor and all transactions are reported to a central analytics cluster that gets all transactions from all Giant Eagle stores. Each store also has separate computers or handheld devices which employees can use to adjust prices and track inventory. The support terminal shows the status of all self-checkout systems to an employee and flashes warnings if any system experiences problems.

End of exam.