# 15-313: Foundations of Software Engineering

Fall 2016 Midterm Exam

Christian Kästner and Claire Le Goues

Name: _____

Andrew ID: _____

## Instructions:

- Not including this cover sheet, your exam should have 8 pages in total: 7 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 detach it and read it first. The appendix will not be considered in grading; 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 4 mulit-part questions with a maximum score of 62 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 with notes. You may not use books, a calculator, cell phone, laptop, or any other electronic or wireless device.
- Good luck!

| Question | Points | Score |
|---|---|---|
| Planning | 14 | |
| Architecture | 20 | |
| Metrics and QA | 10 | |
| Requirements | 18 | |
| Total: | 62 | |

## Question 1: Planning (14 points)

(a) (4 points) *(1 NN)* Name an important *process risk* in developing the system described in the scenario that could delay delivery, and suggest a mitigation strategy.

i. Risk:

ii. Mitigation:

(b) (4 points) *(1 NN)* The "Display Update" service (dashed box in the architecture diagram) looks up, creates, and then sends data to the displays at the various bus stops. This service runs in its own process, querying APIs for the current schedule, estimated arrival times, and messages as needed. It renders the display as bitmap files that it sends to a separate service (REST API) that takes care of the delivery to the physical displays. The colleague in charge of implementing the "Display Update" service says that the implementation will take 5 days, but you're not sure this estimate is trustworthy. To avoid surprise delays, you ask your colleague to define one intermediate milestone.

Give one example of a good milestone for the Display Update service:

*Writing below this line is permitted but discouraged.*

(c) (3 points) *(1 NN)* The head of the operations team (a hardware person) thinks that software is always too buggy and wants to incentivize high-quality code by measuring code complexity (length of the code, number of branches, comment density) and then providing a bonus to the "best" programmer in each team. Do you think this is a good idea? Why/why not?

(d) (3 points) *(1 NN)* Suggest an alternative way to use software analytics (measurement) to increase software quality.

---

*Writing below this line is permitted but discouraged.*

## Question 2: Architecture (20 points)

All architecture questions refer to the architecture diagram shown as part of the scenario, in the Appendix.

(a) (2 points) *(0 NN)* Which architectural view(s) does the diagram use?

(b) (6 points) *(2 NN, must leave both parts blank)* Customers get pretty annoyed when displays show busses that have already passed the stop as "arriving soon." Hence, you are worried about whether your system can get the bus positions and arrival times to the displays before they are outdated. You evaluate whether this is a problem for your system and consider mitigation strategies, such as always computing arrival times that are 30 seconds too optimistic.

i. What insights (if any) does the architectural diagram provide that can help you make this decision?

ii. What additional information would you want to collect to make the decision, and how would you collect it?

(c) (6 points) *(2 NN)* A colleague browses through an architecture book asks you whether it might be useful to introduce the ping-echo tactic[1] between the Display Update service and the physical display components at the bus stops, such that the display could turn off if the Display Update service cannot be reached. The additional calls are shown as dotted lines (for one display) in the architectural diagram. Do you think this would be useful change to the architecture, and why/why not?

(d) (6 points) *(2 NN, must leave all parts blank)* You consider hiring an external consultant to perform an architecture evaluation (eg. ATAM). Provide one argument for and one argument against such an evaluation that is plausible in the context of the scenario. What would you decide and why?

i. Argument for:

ii. Argument against:

iii. Decision (with justification):

[1] One component issues a ping and expects to receive back an echo from the component under scrutiny within a predefined time window. This can be used in a hierarchy to reduce bandwidth.

## Question 3: Metrics and QA (10 points)

(a) (4 points) *(1 NN, must leave both parts blank)* For each of the following quality attributes, give a good metric that you can use to determine if the implemented bus system is satisfying the associated quality requirement:

i. *Usability of the displays:*

ii. *Accuracy of the displayed times as estimated by the system:*

(b) (6 points) *(2 NN, 1 per part)* Your team is trying to decide how to test whether your GPS-sign-updated system is sufficiently accurate, especially in the face of unexpected bus detours. One engineer argues for building a bus simulator and simulating many possible perturbations to a normally running system; another says you should just deploy the system, collect real-time measurements on the live system once it is operational. What are the tradeoffs, in terms of arguments for each, for each of these options?

i. *Build a simulator:*

ii. *Collect measurements on the deployed system:*

## Question 4: Requirements (18 points)

(a) (6 points) *(2 NN, must leave all parts blank)* You have scheduled a requirements elicitation interview with the head of PAT's bus division. You decide to prepare some material to support the interview. For each, explain whether you would or would not want to use it in the interview, and why/why not:

i. A fully-dressed use case

ii. A set of 5 open-ended questions

iii. A walkthrough of the current system deployed in the city of Philadelphia, which is similar but far inferior to the system you plan to build.

(b) (4 points) *(1 NN)* One of your engineers argues that it's completely reasonable to reference the details of the implementation of Philadelphia's system in the use cases you write for your new system, because it provides helpful context and starting points for the development team. Do you agree? Why or why not?

(c) (8 points) *(2 NN, 1 per subpart)* Before our subordinates get down to coding, your manager asks for (A) fully dressed use cases and (B) a fault tree analysis of all risks in the system. Give an argument for and against the use of each in this scenario.

i. *Fully dressed use cases.*
**Argument in favor:**

**Argument against:**

ii. *Fault tree analysis.*
**Argument in favor:**

**Argument against:**

# A Appendix: Scenario description

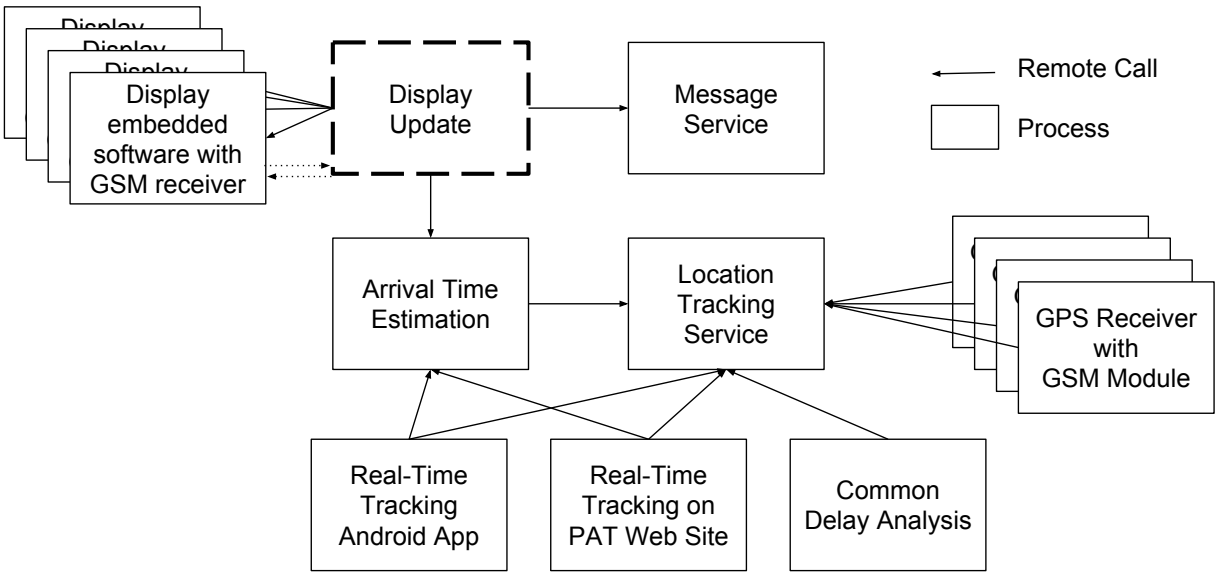
*The following fictional scenario will be the basis of most questions within this exam. Assume that this system is new and that no such system exists in Pittsburgh already (even though it does).*


The Port Authority (PAT), responsible for the busses in Pittsburgh, has finally decided that it should provide more accurate information about bus delays and arrival times. PAT has received a grant for installing displays at 150 of their 8000 bus stops. Each display consists of a pixel matrix that can be updated by a remote procedure call over a GSM (cell phone) network.

The idea is to install GPS sensors and connectivity in each of the 750 PAT busses which then regularly send each bus's position to a central server. The server will determine where the bus is on its route, and then predict updated arrival times. At each equipped bus stop, the display will list the upcoming busses and their predicted arrival times. The same displays might be used to show additional messages, like "LETS GO STEELERS" on football game days.

You are part of the PAT's IT team, a team of 6 developers and 2 IT technicians. In the past, your team was responsible mostly for maintaining the PAT web site (publishing schedules, trip planning, events, ...) and supporting office PCs. While the grant will pay for the new hardware, PAT cannot invest much for the software component and won't be able to outsource development, nor hire new developers. The PAT hopes that two members of your team can dedicate most of their time moving forward to writing the software for this new system.

You have sketched the following architectural diagram of the system. It describes how the on-board GPS modules send information to a central service that stores all current and historic bus locations; that a separate service uses that data to estimate arrival times; and that a display service computes the information to show on each display, taking into account possible messages provided by the message service. Other applications, like the web page, mobile apps, and analytics software might use the same data in the future.




End of exam.