Final Review

Review slides, notes, book, and lab materials. This is a comprehensive exam over the entire semester, including Requirements lecture.
Exam will be closed book. However, as a good study practice, it is suggested you prepare a one page summary of key definitions and terms in order to help you learn and practice the material.

The following is an example exam.

Multiple Choice

1) Which of the following is *NOT* an example of a requirement solicitation technique?

- A) Interviews
- B) Observation
- C) Examining Documents and Artifacts
- D) Focus Groups
- E) None of the above

2) What HTTP verb would be most appropriate for editing an entity in a RESTful service?

- A) PUSH.
- B) PATCH.
- C) PUT.
- D) B or C
- E) None of the Above

3) In requirements engineering, which of the following is *NOT* an example of a non-functional requirement or constraint?

- A) The unit tests must be written in JUnit.
- B) The product shall not be larger than 56 MB when installed.
- C) The product shall not store passwords in the database.
- D) The product shall allow a doctor to reset a password.
- E) None of the above

4) Which of the following best describes encapsulation?

- A) Measure of interdependence between two objects
- B) When methods declared in one class use methods or attributes of the other class
- C) Number of direct descendants (subclasses) for each class
- D) Measure of the proportion of attributes that are “invisible” from other classes or objects
- E) Degree to which the tasks performed by a single module are functionally related

6) Which deployment practice definition best describes a dark launch?

- A) Release software without any user-facing elements exposed, use flags to turn on in production.
- B) Start with two identical instances of infrastructure. Deploy beta version on one, stable on other. Switch when ready.
- C) Stop traffic to node, upgrade node, reroute traffic back, upgrade next node.
- D) Sample a small amount of traffic, eventually route more traffic if stable.
- E) B or D

7) Which is the following is a disadvantage of big switch deployment?

- A) May be supporting multiple versions in production.
- B) Cannot simulate the impact of many real users
- C) Rolling back is easy
- D) A and C
- E) None of the above

8) Which of the following best supports building confidence in a proposed deployment candidate in the context of a production-like environment.

- A) staging

- B) dark launches
- C) canaries
- D) release planning
- E) mocking

9) What lesson did Netflix learn after deploying a Chaos Gorilla?

- A) Startup resiliency is often missed
- B) Fallbacks can fail too
- C) Infrastructure control can be a bottleneck
- D) Label everything (do we need this?)
- E) State is bad

Short Answer

1. Describe the principle of "Cost of Change is Dead"

2. Explain the difference between docker and ansible.

3. A manager asks to replace all try/catch code with a try-with-resources.
   What style of refactoring is this?

4. Name 2 ways user input can cause security issues. What are some measures that help?

5. Provide an example of a requirement. Provide an example of a non-functional requirement.

6. Describe the principle of "Rule of 3".

Scenarios

1. A company wants to increase the responsibility of their development team to include handling testing, staging, and production. They want to reduce communication barriers and team silos separate groups imposes. What team model might be appropriate?

2. A company has recently adopted continuous deployment of their product, but has started receiving complaints from their customer that new features are coming too fast and running too slow. What practice would you recommend they adopt to prevent these problems while allowing continuous deployment?

Implementation

1) Measure the branch coverage of a code snippet. (10 points)

Test suite:

- `weird(0, 0, 0, "strictly", [0,0])`
- `weird(88, 42, 42, "stricter", [0,0])`

```
function weird(x,y,z, mode, results)
{
      if( x > 100 || y > 70 )
      {
          z = 33;
          results[0] = getData().data.server().first.name;
      }
      else if( z == 0 && y < 0)
      {
          if( mode == "strictly" )
          {
              return 0;
          }
      }
      else if( z < 42 )
      {
          if( mode != "stricter" )
          {
              results[1] = getData().cachedResults.first.name;
              return y = z / x;
          }
      }
      return 1;
  }
```

2) Write a parser using a simple visitor pattern.

You can assume a basic visitor function already exists and provide your own tokens and AST structure you assume exist (feasibly parsed).

A) Count the number of branches in a function. (10 points)

- What is number of branches in the above weird function?
- Parser implementation.

B) Count the number of conditions in a function. (10 points)

- What is number of conditions in the above weird function?
- Parser implementation.

Analyze

1) Describe two advantages and two disadvantages of crowd documentation. Describe one way in which a software company may try integrating crowd documentation in development of an API?

2) Compare and contrast using mocking versus canary releasing. Describe their impact on testing.

Design and Architecture

1) Use class diagrams to illustrate how your team supported PT and Ortho office visits in iTrust.

2) How might you create a microservice architecture of iTrust, running in the cloud? *Focus on creating a well annotated diagram that illustrates essential components, paragraphs of just text will not be given credit.*