University of Illinois at Urbana-Champaign
Department of Computer Science

Final Exam

CS 428 Software Engineering II
Spring 2013

May 7, 2013

TIME LIMIT = 3 hours
COVER PAGE + 20 PAGES

Upon receiving your exam, print your name and netid neatly in the space provided below; print your netid in the upper right corner of every page.

Name: ______________________________
Netid: ______________________________

This is a closed book, closed notes examination. You may not use calculators or any other electronic devices. Any sort of cheating on the examination will result in a zero grade.

We cannot give any clarifications about the exam questions during the test. If you are unsure of the meaning of a specific question, write down your assumptions and proceed to answer the question on that basis.

Answering **I don't know** (and nothing else) to any exam question is automatically worth 25% partial credit for that question.

Do all the problems in this booklet. Do your work inside this booklet, using the back of pages if needed. The problems are of varying degrees of difficulty so please pace yourself carefully, and answer the questions in the order which best suits you. Answers to essay-type questions should be as brief as possible. **If the grader cannot understand your handwriting you may get 0 points**.

There are 13 questions on this exam and the maximum grade is **126 points**.

Page:	1	2	3	4	5	6	6	7	8	9	10	Total
Points:	10	10	8	8	5	2	6	7	4	11	2	73
Score:												

Page:	12	12	14	15	16	19	20	Total
Points:	3	5	9	12	8	8	8	53
Score:								

Your final score in this exam is: __________

1. Project Management & Risks

3 (a) You work on a mobile application. You are part of a team of 4. For the current project, only one member, Alan, knows the complete specifications and the core system. Think about one risk that can affect the project. Also, provide the risks type, and describe a way to mitigate it.

Solution:
One of the risks is that Alan might leave the team. A way to mitigate is to have other team members pair program with him to learn the system. Also, you could write the specification in a document for further reference.

3 (b) In your class project you had to perform risk management. Describe three activities involved in risk management.

Solution:
Risk management mainly consists of:

- risk detection and classification (severity, probability)
- defining and updating mitigation plans for identified risks
- periodical risk monitoring. Risks change in severity and probability over time. Mitigation plans change over time

Projects usually have a risk table that contains above mentioned risks.

3 (c) Describe how the cost of change relates to the phase of the project. Then explain two advantages of a highly iterative process from this point of view.

Solution:
The cost of defects grows exponentially from requirements to release. The costs of changing reqs grows exponentially in the same manner.

By using a frequent release process, this req-design-code-test release cycle is shortened to the length of an iteration. Therefore any defects and requirements change that creep in mostly affect the an iteration, thus it is less painful to recover.

Moreover, a highly iterative process enables constant validation and re-alignment of requirements by ensuring that at the end of each iteration there is a functional, partially complete system with which the client can evaluate.

2. Requirements

1 (a) In the interview, Alistair Cockburn stated that business people and developers should work together when writing use cases. Why?

Solution:

Business people tend to write use cases very vague and ambiguous. Developers tend to go on the other side. By having them work together, they can come to a common language that both of them can understand.

1 (b) What is wrong with the following use case?

> When the user clicks the checkout button a new page appears with the details of his delivery. After he fills it in, he clicks on the left button and a confirmation dialog appears asking him to confirm his purchase.

Solution:
The use case is written in terms of the GUI

3. Component Design

2 (a) What is the most common misuse of the layered architecture? Why is this a problem? Hint: There was a discussion in the interview with Dragos Manolescu.

Solution:
It is very tempting to bypass the layerings. You do not want to go from the user interface directly to the database.

2 (b) Parnas and Clemens argue that we need to fake the design process. Why can't it be done right the first time?

Solution:
Requirements change during the development process. Understanding all the details before we starting developing is almost impossible. During a project, we are bound to make errors. These errors will force us to change our design over time. Sometimes, we have to cut corners. (one argument will suffice).

4. Architecture

3 (a) Consider the architecture of a computer: transistors, logic gates, registers, counters, processors, instruction sets, assembly, operating systems (kernel, system call interface, etc), web services, etc. What type of architecture is this? Write two characteristics of this architectural style.

Solution:
Layered. Each layer abstracts away some detail and provides a stable abstraction over it. Transistors abstract away electronic physics, gates abstract away boolean logic, counters abstract away addition, etc. Each layer does its job by depending on the abstraction below it and in its turn encapsulates some piece of complexity from the layers above which are relieved of that particular burden.

2 (b) What is the main difference between the traditional MVC and the web MVC?

Solution:
No "observer" relationship between the view and the database (passive implementation)

3 (c) You have to implement a system that tries to collect all knowledge about a person in one big graph. There are many types of information that can be collected or inferred about a person and each requires its own trait processor (spiders or analyzers). Spiders crawl the web and bring raw data (artistic preferences, financial data, product preferences, travel history etc) into the graph of knowledge. Analyzers study the raw data to infer higher level traits about that person (diverse psychologic profile fragments, future purchase trends, future travel trends, etc) and then reinforce the graph with their findings. Many processors lie dormant until the graph acquires certain properties that enables them to perform their analysis. There are hundreds of processors that bring, reinforce and infer new data from and into the graph at random points in time. It is certain that in the future many other types of such processors will be added.

Here is a list of potential architectural patterns: Central Graph, Observer, Layered, Scraper, Umbrella, Repository, Abstract, Weaver, Pipes and Filters, Aggregator, Client - Server, Composite, Crawler, Flyweight, Analyser.

Which one best fits the above system? Why?

Solution:
Repository / Blackboard pattern. There is a central repository of knowledge. There are many specialized agents that add or consume certain parts of repository. The agents work together to solve a bigger problem. Each agent is specialized on a particular task thus it waits until the repository has the certain features that enable it to perform its task.

3 (d) The main design principle in MVC is the separation between the presentation layer and the problem domain model. Name three reasons why it is important to have the two layers separated.

Solution:

- they have different reasons and rates of change
- they need to be able to evolve separately
- easier to test

2 (e) When you write bash commands you might use something like this:

```
ls -l | grep *.java | sort
```

if you want to show all java files ordered by size. What architectural pattern does it remind you of? Why?

Solution:
Pipes and filters. Data is processed sequentially through a series of commands. It is “piped” in between using the pipe operator.

2 (f) How is architecture decided in XP?

Solution:
In XP there is no predetermined architecture. The architecture is “grown” during the development process. A good developer will start with the important user stories first to get to a good architecture.

5. Specification and Abstraction

3 (a) Are tests suitable to express specifications? In your answer provide three arguments.

Solution:

- easy to see if program meets specification
- easy to understand both specification and implementation
- forces you to think about api design
- can be implemented in the same language as system language → require no other language

3 (b) The following code is an example of Abstraction Inversion anti-pattern. Describe the anti-pattern. Why is it bad? How would you fix it? Please use the example to illustrate your point.

```java
class Person {
  private List<Integer> age = new ArrayList<Integer> ();

  public Person(int age){
    this.age.set(0, age);
  }

  public int getAge(){
    return age.get(0);
  }
}
```

Solution:
The antipattern is Abstraction Inversion because it uses an ArrayList to represent an Integer. It uses a high level concept to implement a low level concept. It impacts performance and readability of the code.

5 (c) Read the code below. From the point of view of encapsulation and abstraction, identify three things that are wrong with this code. Explain what changes you would make in order to improve the abstraction and encapsulation of this code. You can use plain English, pseudocode or UML diagrams to make your point.

```
 class Employee {
  public static final int HOURLY_RATE_EMP = 0;
  public static final int FIXED_RATE_EMP = 1;
  public static final int SALES_EMP = 2;

  public String name;
  public int type;
  public float hourlyRate;
  public float hoursWorked;
  public float fixedSalary;
  public float salesPercentage;
  public float salesAmmount;

  public Employee(int type) {this.type = type;}
}

class SalaryUtilities {
  public void calculateSalaries(List<Employee> employees) {
    for (Employee e : employees)
      if (e.type == HOURLY_RATE_EMP )
        System.out.println(e.name + ": " + e.hourlyRate * e.hoursWorked);
      else if (e.type == FIXED_RATE_EMP )
        System.out.println(e.name + ": " + e.fixedSalary);
      else if (e.type == SALES_EMP ) {
        double salary = e.fixedSalary + e.salesPercentage * e.salesAmmount;
        System.out.println(e.name + ": " + salary);
      }
  }

  public float maxSalary(List<Employee> employees) {
    float max = 0;

    for (Employee e : employees) {
      float salary = 0;
      if (e.type == HOURLY_RATE_EMP )
        salary =  e.hourlyRate * e.hoursWorked;
      else if (e.type == FIXED_RATE_EMP )
        salary = e.fixedSalary;
      else if (e.type == SALES_EMP )
        salary = e.fixedSalary + e.salesPercentage * e.salesAmmount;

      if(salary > max) max = salary;
    }
    return max;
  }
}
```

Solution:
Data of employee abstraction not tied with its behaviour (lack of encapsulation). Code duplication in client code. Hard to change logic for employee behaviour because it is

spread out. Extract three subclasses from Employee: fixed rate, hourly rate and sales based. Introduce computeSalary method in superclass. Use computeSalary polymorphically in client code.

6. Quality Assurance

2 (a) One of the main benefits of code reviews is finding defects. List two other benefits of doing code reviews.

Solution:
With enough eyeballs all bugs are shallow. Suggest code improvements, transfer programming knowledge, suggest alternate solutions, increase ownership of the code

2 (b) Read this statement: "External characteristics of quality are the only kind of software characteristics that users care about. Therefore programmers should focus only on external quality characteristics throughout the development process".

It this true or false? Explain your answer.

Solution:
False. Although external quality characteristics are indeed the only software characteristics that users care about, the programmers focus should be on the internal quality characteristics since the internal ones provide long term support for, and in many cases induces, external qualities. With time, unkept software rots and it becomes hard to change, error prone, hard to reuse and hard to comprehend. This in turn brings a halt to the external quality characteristics since it would require tremendous costs to keep and extend them.

Or true if the system does not need to change afterwards.

2 (c) The manager wants the highest possible defect detection rate. Which quality control techniques should he use? Why?

Solution:
He should use a combination of techniques since not one of them provides adequate detection rate in isolation .

7. Low Level Design

2 (a) Bill Gates remarks that measuring a program by the number of lines of code is like measuring an airplane by how much it weighs. Do you agree? Argue your position.

Solution:
Yes, he is correct. The number of lines of code does not tell us anything about the design, complexity, etc.

5 (b) For the following problem, fill the provided decision table. Your solution **must** fit in the provided table.

> A customer gets free shipping if she spends over $50 and requests domestic delivery. If the delivery is international, she must spend $100 or more. Additionally, a customer who spends $200 or more gets free shipping and a 5% discount on his items, regardless of the shipping address. If he spends $300 or more she gets free shipping and a 10% discount.

Rules Conditions	1	2	3	4	5	6
Spends between $50 and $99	Y	N	N	N	N	N
Spends between $100 and $199	N	Y	N	N	N	N
Spends between $200 and $299	N	N	Y	Y	N	N
Spends over $300	N	N	N	N	Y	Y
Domestic shipping	Y	N	Y	N	Y	N
Internation shipping	N	Y	N	Y	N	Y
Actions						
Free shipping	X	X	X	X	X	X
5% discount			X	X		
10% discount					X	X

2 8. Uncle Bob talked about the advantage that Object Oriented Programming has over Abstract Data Types. What is this advantage?

Solution:

The ability to independently deploy the components/modules of our system. This also means that our system has low coupling, so the design is better.

9. The following is a piece of Java code modified from `BatchCommitLogExecutorService` class in the Apache Cassandra project:

```
class BatchCommitLogExecutorService {

  private final ConcurrentLinkedQueue<CheaterFutureTask> queue;

  private ArrayList<CheaterFutureTask> processWithSyncBatch(){
    ArrayList<CheaterFutureTask> incompleteTasks =
      new ArrayList<CheaterFutureTask >();
    while (!queue.isEmpty()) {
      CheaterFutureTask task = queue.poll();
      incompleteTasks.add(task);
      taskValues.add(task.getRawCallable().call());
    }
    return incompleteTasks;
  }
}
```

The method `processWithSyncBatch` is supposed to be executed CONCURRENTLY by two threads, so the code uses a concurrent data structure `ConcurrentLinkedQueue` from the Java concurrent library. Please read the code and answer the following questions:

3 (a) This code contains a bug due to concurrent execution. What is the result of the bug and how can it happen? Show the thread interleaving that causes the bug. (hint: based on JavaDoc, `ConcurrentLinkedQueue` is thread-safe and the semantics of `queue.poll()` is “retrieves and removes the head of this queue, or returns null if this queue is empty”)

Solution:
There is one element in the queue, both threads make the empty check and only one gets the element. The other one gets null and will cause a null pointer exception.

1 (b) What is the name of this kind of concurrent bug pattern?

A Data race
B Atomicity Violation
C Order Violation
D Deadlock

Solution:
B

10. User Interface

2 (a) The book Designing Interfaces by Jenifer Tidwell recommends One-Window Drilldown for designing an interface for a device with a small screen (like a mobile telephone). Explain why.

> **Solution:**
> The application consists of many pages of content that the user must navigate through. Because its simple and does not require a lot of screen space it is very suitable for small screen displays. Also, the simple model is easy for users to understand (think of a browser).

2 (b) Tog presents the concept of the mile high menu bar. Please explain the UI design principle supporting the concept behind it, and how it affects the mile high menu bar.

> **Solution:**
> Users have difficulty in precise mouse adjustment. Therefore, UI elements must be placed in way that require the least amount of precise movement. By placing the Mac menu bar at the top of the screen the users dont need to have precise movement on the Y axis anymore. They only need to worry about positioning the cursor horizontally.

2 (c) What do the following entities afford? When designing User Interfaces, why should you include entities with specific affordance?

1. Door knob
2. 3D button
3. Scissors
4. Chair

> **Solution:**
>
> 1. Door knob - twisting
> 2. 3D button - pushing / clickable
> 3. Scissors - cutting
> 4. Chair - sit
>
> It gives the user a strong hint on how the UI should be used.

2 (d) In the interview with Joel Spolsky he mentioned the Amazon 1-click order as an example of a great user interface. Please provide two reasons.

> **Solution:**

3 (e) Joel Spolsky states that interfaces should require minimum memory from users. For example, mobile phones allow a user to assign names to phone numbers because names are easier to remember. Consider these scenarios:

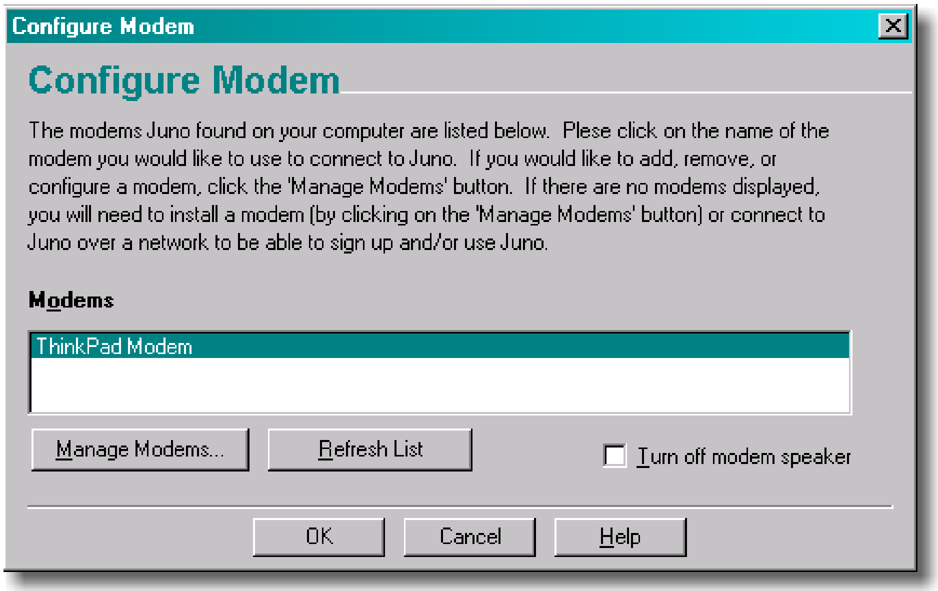
1. The user is prompted to select a photograph for his avatar and is shown a list of file paths to the photos.
2. The user wants to load a file and is required to input the file path.
3. The user wants to edit a photo via a series of adjustments. For increasing the saturation he needs to type in the “bumpSat” command. For adjusting the contrast he needs to type in “contrastScan [options]” command. For adding a gradient he needs to type in the “invokeRenderer [options]” command. So on and so forth.

How do the above three scenarios break the minimum memory principle? Describe how would you change them to follow the principle.

Solution:

1. Use thumbnails instead of filenames. User does not need to remember photo - filename associations.
2. Use a file browser to locate the file. User can explore the file system and does need to remember specific filepaths.
3. Mark the adjustments graphically through the use of icon panels or menus. The user does not need to remember specific commands.

2 (f) What is the main problem with the following window? Why?



Configure Modem

The modems Juno found on your computer are listed below. Plese click on the name of the modem you would like to use to connect to Juno. If you would like to add, remove, or configure a modem, click the 'Manage Modems' button. If there are no modems displayed, you will need to install a modem (by clicking on the 'Manage Modems' button) or connect to Juno over a network to be able to sign up and/or use Juno.

Modems

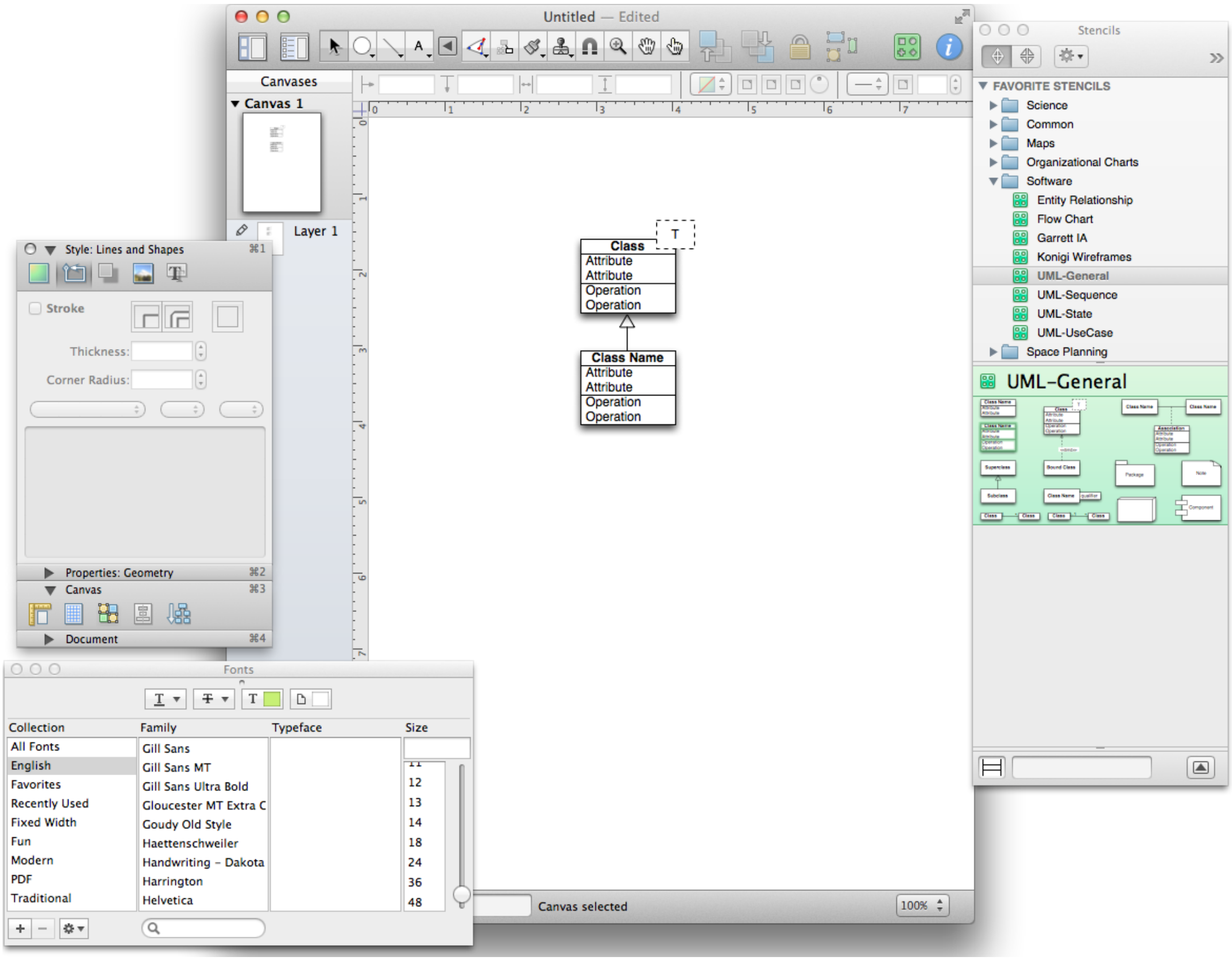
ThinkPad Modem

Solution:
Too much text. It is useless for experienced users and novices will either skip it or get

confused by it.

3 (g) The figure below is a screenshot of Omnigraffle Professional. Select **three** User Interface patterns you can identify in the figure, from this list of possible UI patterns: Visual Framework, Model View Controller, Composite Windows, Center Stage, Layout Interpreter, Liquid Layout, Responsive Disclosure, Decorator, Movable Windows, Front Facade, Progress Indicator, Cascading Strategy, Card Stack, Mirror, Titled Sections, Dropdown Chooser, Layouter, Sketch Cards, Closable Panels, Movable Panels, Liquid Layout, Fractal Views, Tree-Table, Smart Selection, Composite, Wizard, Intriguing Branches, Overlapping Windows, Diagonal Balance, Observer Link, Custom Menus.

Also, please circle/mark them on the figure.

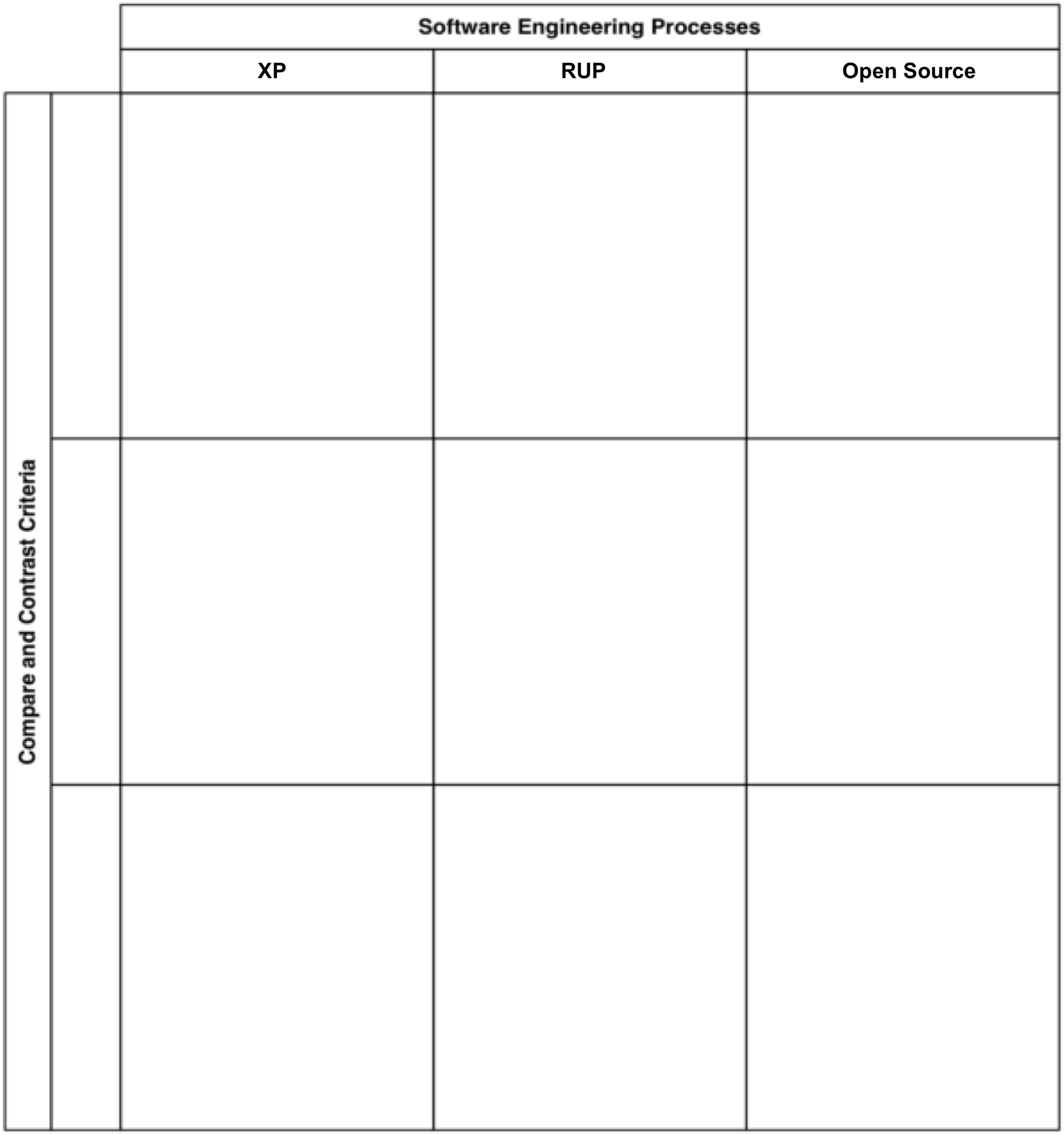


Solution:

11. Processes

5 (a) In CS428 you have learned a number of different agile development processes, some of whom which you have followed in your project. Different processes have different characteristics, strengths and weaknesses. In the table given below, compare and contrast XP, RUP and Open Source Development for **three** categories. Select the categories from the

ones listed below: **Testing, Roles, Activities, Values, Planning, Risk Management, Deliverables, Design and Architecture**



		Software Engineering Processes		
		XP	RUP	Open Source
Compare and Contrast Criteria				

12. Open Source

1 (a) In response to “The Cathedral and Bazaar”, Fred Brooks stated that “It works when the builders are the clients”. Please comment on his statement

Solution:
They know the requirements from personal experience. The need of having the system is their own need.

2 (b) Open Source Software has been compared to a bazaar with a small, core cathedral in the middle. Please explain this comparison keeping in mind what you read in “The Cathedral and the Bazaar”

Solution:

- big software design decisions cannot be made by the crowd. A small core makes the core, minimal grand design and then the OSS community fills in the holes.
- design ideas are contributed by the whole community but the cathedral ultimately decides. Design ideas bubble up from the bazaar up to the cathedral.
- everybody can contribute to the code base but there is a hierarchy of gatekeepers who ultimately decide on what goes in. Ex: Linux, Eclipse, etc

13. You are working at a company that builds software solutions for small and medium sized hotels. Your team is responsible for the backend part, where you have to handle the business logic. You have to implement the following scenario:

> The system must allow users to log into the system and reserve a room. Some hotels also allow you to reserve special items (like skis, snowboards, bikes etc). These are added to the room reservation and show on the bill as separate items. The customer can then change her reservation or cancel it before the deadline. Cancellation and change depends on a cancellation policy. At the end of the stay, the customer receives an itemized bill for the services she used.
>
> Each hotel can have different options of extra items, so the manager must be able to manage these. Some of the extra items may be available year round, while some may be available only during a certain time frame (e.g., you can only rent skis during winter).

6 (a) Using the brief use case format, write the use cases for the above system.

Solution:

1. User creates a reservation
2. User changes a reservation

3. User cancels a reservation
4. (1-3) User manages a reservation
5. Receptionist bills a customer
6. Manager adds/edits/removes (manages) extras and the time when they are available.

4 (b) For the above system draw a use case diagram.

Solution:

8 (c) For of the "Bill a Customer" use case, please expand it into a fully dressed use case. Fill in the provided template.

Use Case:

Primary Actor:

Goal in Context:

Scope:

Level:

Stakeholders and Interests:

Precondition:

Minimal guarantees:

Trigger:

Main Success Scenario:

Extensions:

8 (d) For the system, draw CRC cards corresponding to it. Use the ones drawn on the next pages. If you need more, draw them on the back side of the pages.

8 (e) Draw a class diagram of the system.

Solution:

8 (f) Draw the sequence diagram for the ”Bill the customer” use case:

Solution: