

## CECS 343 — SWE Intro — Exam #2 – Answers

Page Name	Type	Points	Diff	B
1 Q1. What's viable about an MVP?	SA	10	1	
2 Q2. How does RPC differ from just calling a function in ...	SA	10	1	
3 Q3. How does an ORB component help lower coupling in a S...	SA	10	1	
4 Q4. How does a Buffered architectural pattern help lower...	SA	10	1	
5 Q5. When the users get their hands on a program, there a...	SA	10	1	
6 Q6. What does the acronym BUFD stand for?	SA	10	1	
7 Q7. In Brooks' view of incremental S/W development as be...	SA	10	1	
8 Q8. Very briefly, what is difference in purpose of the A...	SA	10	1	
9 Q9. Very briefly, what is the main structural difference...	SA	10	1	
10 Q10. Very briefly, what is the main structural difference...	SA	10	1	
11 Q11. What does the 'U' in FURPS stand for?	SA	5	1	
11 Q12. What is the Cyclomatic Complexity of this piece of c...	SA	10	1	
12 Q13. Very briefly, what does regression testing guard aga...	SA	10	1	
13 Q14. Very briefly, what is the major reason for at least ...	SA	10	1	
14 Q15. As discussed in lecture, there are several major way...	SA	10	1	
15 Q16. As discussed in lecture, there are several major way...	SA	10	1	
16 Q17. Before the contract is signed in a ("big bang" style...	SA	10	1	
17 Q18. Sometimes there are places within the S/W project co...	SA	10	1	
18 Q19. What does the acronym SDLC stand for?	SA	5	1	
19 Q20. What Agile value emphasizes extensive discussions wi...	SA	10	1	
20 Q21. For the sequence from discussing with the users how ...	SA	10	1	
21 Q22. What is the main reason given for keeping Agile team...	SA	10	1	
22 Q23. What is the main difference between a Sprint Backlog...	SA	10	1	
( 23 Q24. What is the most important reason for performing a d...	SA	10	1	3
24 Q25. The five major reasons that Michele Jackman listed i...	SA	10	1	
Total		240		

## CECS 343 — SWE Intro — Exam #2 – Answers

Curve denominator = 215

Class GPA 2.26

### Q1

. What's viable about an MVP?

**A: It's sales (money) will support itself (its continued upgrades)**

7pt = Acceptable product that customer will buy/pay-for

2pt = You avoid spending time building the wrong stuff

--

### Q2

. How does RPC differ from just calling a function in a program?

**A: The fcn called by RPC resides on diff (remote, usually) PUs**

9pt = RPC sends reqs to a Server

9pt = RPC requests a function call from another program

3pt = RPC is Blocking calls until it receives an answer

—

### Q3

. How does an ORB component help **lower coupling** in a S/W program?

**A: It helps ensure that Requesters & Services don't know each other.**

8pt = Connects the Requesters to the Services // important

8pt = Acts as a middle man

5pt = Acts as a Broker // part of ORB name

--

## CECS 343 — SWE Intro — Exam #2 – Answers

### Q4

. How does a Buffered architectural pattern help lower coupling in a S/W program?

**A: Each agent can't talk directly to (ie, call) its neighbor, and instead leaves a message in a data store/buffer.**

5pt = Can change either agent easily // why? don't comm directly

5pt = Data-storage "buffer" between two agents for communication

--

### Q5

. When the users get their hands on a program, there are three prominent things that are typically discovered about the initial set of requirements, per Brooks. Very briefly, name two of these three things.

A: 1) devrs find mis-understandings;

2) users realize poor reqts;

3) users imagine new better reqts

3pt = model will turn out to be different than how initially intended

--

### Q6

. What does the acronym BUFD stand for?

A: Big Up-Front Design

--

## CECS 343 — SWE Intro — Exam #2 – Answers

### Q7

. In Brooks' view of incremental S/W development as being a **significant morale booster** for the development team, at what stage in the S/W development process is the software program expected to work?

**A: "One always has, at every stage in the process, a working system."**

3pt = Quick prototype for the users

2pt = Before the build phase – (before or at ANY particular “phase”)

2pt = Before deployment with the Testing stage uses V&V

(\*) When somebody says "Morale", you say "Yes"

Idea:

- Brooks (Turing) says Iterative Incremental Delivery wins (30yrs ago)
- Weinberg says very short (**half-day**) iterations, with demo (54yrs ago)
  - o- planning and writing tests before each **micro**-increment (= EIO)
- Agile says Mini-Incremental Dev & Delivery
  - o- TDD = write tests before design/code (= EIO)
- DevOps CI says Daily/Continuous build+regression\_test of every checkin
- Rule #2 (Hunt) says small Add-a-Trick & see visible success (focus on 0 run-time bugs)
  - o- Rule #4 (EIO) says write tests before design/code (helps focus design & avoid gold-plating)
- Smart Std says to always have visible progress to show (your manager), daily
- Rule #5 (Half-Day) says plan in visible half-day tasks

Get the Picture? --> **Stop writing large/medium chunks of untested code**

—

## CECS 343 — SWE Intro — Exam #2 – Answers

### Q8

. Very briefly, what is difference in purpose of the Adapter GoF design pattern versus the Facade GoF pattern?

**A: Adapter allows classes with incompatible interfaces to work together; whereas Facade Wraps multi-class weird-API group of classes with a single nice-API class.**

5pt = one of them

NB, Adapter != Decorator

—

### Q9

. Very briefly, what is the main **structural** difference of the Decorator GoF design pattern versus the Composite GoF pattern? (as opposed to the behavioral difference)

**A: The Composite is used for trees of objects, whereas the Decorator is used for a list of objects.**

(Tree vs List structures)

9pt = Decorator does before/after actions, Composite has kids array

8pt = Decorator works on one kid, Composite works on all kids

5pt = Decorator has array of IHref slots to next component; // needs only 1

Composite has array to other components // tree

3pt = Composite works on compositions of kids

3pt = Both connect leaf and non-leaf agents

—

## CECS 343 — SWE Intro — Exam #2 – Answers

### Q10

. Very briefly, what is the main **structural** difference of the Document style of very large databases versus the style that just uses keys paired with values?  
(and the 3-rd kind of NoSQL large DB is called a Graph DB)

**A: A Document DB has hierarchically structured data.**

(uses JSON – like XML/HTML – with hierarchical objects containing smaller objects – textually)

7pt = Large databases use Hierarchical Structured Data while the style just uses dictionaries

4pt = Key-value pairs can index directly

[CF [aws.amazon.com/nosql/document/](https://aws.amazon.com/nosql/document/)]

Document Database Defined

A document database is a type of nonrelational database that is designed to store and query data as JSON-like documents. Document databases make it easier for developers to store and query data in a database by using the same document-model format they use in their application code.

The flexible, semi-structured, and hierarchical nature of documents and document databases allows them to evolve with applications' needs.

The document model works well with use cases such as catalogs, user profiles, and content management systems where each document is unique and evolves over time. Document databases enable flexible indexing, powerful ad hoc queries, and analytics over collections of documents.

]

—

### Q11

. What does the 'U' in FURPS stand for?

**A: Usability**

—

## CECS 343 — SWE Intro — Exam #2 – Answers

### Q12

. What is the Cyclomatic Complexity of this piece of code?

```
do {  
    if (a < 3) { x += 2 * a; }  
    else if (b < a) { x += 2 * b; }  
    else { x += 2 * c; return x; };  
    if (x > 100) { return x; }  
    b = a;  
} until ( b > c )
```

**A: 5** // 3 IFs + 1 Loop + 1 ("exterior face") – graph theory style of counting

"Alt: if we count Branch stmts (Ifs and LOOPS) and add 1, we get M"

8pt = 4

6pt = 3

4pt = 2

--

### Q13

. Very briefly, what does regression testing guard against?

**A: That a program's previously working features don't suddenly stop working.**

5pt = Does not produce unintended bugs or breaks in the code

[NB, we care about OLD features/functionality breaking, not about code]

--

## CECS 343 — SWE Intro — Exam #2 – Answers

### Q14

. Very briefly, what is the major reason for at least ensuring **branch coverage** when testing a program?

**A: If you don't, then some executable stmt will have never been tested.**

(Branch coverage means taking both branches of a test. == **all stmts.**)

5pt = To ensure each condition runs at least once

4pt = Because you cannot reasonably test all paths

2pt = To ensure path coverage & show whole program works

2pt = To find bugs

(A “branch” is a THEN body or an ELSE body – a “condition” helps decide which branch you take.)

—

### Q15

. As discussed in lecture, there are several major ways that a S/W project manager can fail, thus causing a project to fail. Name 3 of them.

**A: 1) manage developer morale,**

**2) arrange adequate/effective comm w users,**

**3) see looming risk events in time,**

**4) gel group into a team**

4pt = <one of em>

--



## CECS 343 — SWE Intro — Exam #2 – Answers

### Q16

. As discussed in lecture, there are several **major ways** that can be used to fix (ie, reduce) the **complexity** of S/W on a new-development project. Name 3 of them.

A: 1) Reduce the number of control pathways

2) Reduce the complexity of data flowing between boxes

3) Make some/most/all components “functional”

4) COTS, Reuse Reliable parts

BUT NOT 5) Include 1+ very strong pgmrs on your team

4pt = Don't optimize, don't build for reuse/future

4pt = Use agents not classes, don't build for reuse/future

4pt = SRP from SOLID, Encapsulation, and don't generalize

4pt = High Cohesion, Encapsulation, Low Coupling

3pt = Combine conditionals, break up a function, manage size of methods

--

### Q17

. Before the contract is signed in a ("big bang" style) S/W project, a preliminary analysis is performed to identify, via top-down decomposition, the **likely S/W tasks** that will need to be completed. What is the technical term for this **top-down set of tasks**?

A: WBS == Work Breakdown Structure

8pt = WBS & Feasability estimation // not Feasibility

--

## CECS 343 — SWE Intro — Exam #2 – Answers

### Q18

. Sometimes there are places within the S/W project code where some condition or other is **always supposed to be true**. A) What is the technical term for these conditions, and B) as discussed in lecture, what is a useful coding mechanism that can be used to detect and issue an error if one of these conditions turns out to be false?

A: A) **Code Invariants**, and B) **Assertions**

5pt = one of them

2pt = Conditions & IF stmt

--

### Q19

. What does the acronym SDLC stand for?

A: S/W Development Life Cycle

--

### Q20

. What **Agile value** emphasizes extensive discussions with the users (and customers) **rather than** extensive analysis, detailed creation of, and a give-and-take over those details, for a **legal agreement** to develop and purchase S/W, including various **penalty clauses** in case of failures of various kinds?

A: **Customer collaboration over contract negotiation**

6pt = team/customer oriented- Satisfying the customer is the highest priority

4pt = Satisfy the customer through early and continuous delivery

4pt = Extreme user feedback loop

2pt = <non-Value but Agile>

--

## CECS 343 — SWE Intro — Exam #2 – Answers

### Q21

. For the sequence **from discussing with the users** how they think they will work with a new program **to solve** (or how they are already manually solving) their problem, **to the informal validation** on paper (ie, as a first cut) of a problem domain-level architecture/model -- ideally this sequence should go through **four major steps**, as discussed in lecture. Name those (about four) steps in sequence: what to what to what etc.

A: UScens -> UCs -> CRCs -> CRC hand-sim

2pt = Sequence diagrams ...

--

### Q22

. What is the **main reason** given for **keeping** Agile **teams** rather **small** (eg, a dozen or fewer team members)?

A: Manifesto Item #6 – Because The most efficient and effective method of conveying information to and within a development team is **face-to-face conversation**.

3pt = Make quicker decisions develop team faster, more efficient planning

2pt = Gel as a team better

1pt = Greater transparency increases trust among team members

1pt = Create significant autonomy // Only if Mgmt allows

[WRT "Small teams provide greater transparency and increase trust".

Re “increase trust”:

Then why do some small teams have "free rider" problems – members who don't do significant work?

]

--

## CECS 343 — SWE Intro — Exam #2 – Answers

### Q23

. What is the main difference between a Sprint Backlog and a Product Backlog?

A: The Sprint Backlog is only what is planned for the next delivery (ie, at the end of the current time-box (AKA Sprint))

8pt = Sprint prioritizes for the next sprint while product prioritizes the top 6

7pt = Product backlog is the top 6-ish stuff that consumers want; print backlog are the items taken from the product backlog to put into the next sprint

6pt = Product Backlog is list of product features; Sprint Backlog is list of sprints in action working on feature

6pt = Product backlog is the list of product features; sprint backlog is the time it takes to complete a product feature

5pt = Product Backlog is the start (first 6-ish) of each sprint, Sprint backlog is for the next Sprint

5pt = product backlog is a list of product features; sprint backlog is a list of sprints in action

[ Product Backlog = all features user wants (currently) (eg, 67 items – a large pile of em)  
– and, only top (say) half dozen are prioritized – cuz pry-s will change & don't waste effort

Sprint Backlog = the Product Backlog “features” to be done in the Sprint we are about to do  
– these items (as “user stories”) are in the Ready column of the Progress Board  
– Also, the users help choose these items

]

--

## CECS 343 — SWE Intro — Exam #2 – Answers

### Q24

. What is the most important reason for performing a demonstration of the about-to-be-delivered project upgrade after the most recent Sprint (in the Scrum style of Agile)?

A: Important as a Morale boost for the team // And was also called out in Brooks' Silver Bullet paper, 5.3 Incremental Dev

5pt = make sure the team is building the right product

5pt = demoed to the users for feedback

4pt = to the PO for feedback

3pt = make sure it works

3pt = to improve process for the next project

--

### Q25

. The five major reasons that Michele Jackman listed in her Team Toxicity paper relate to which of the four major causes of project failure?

A: Bad Managers

[the 4 majors are Complexity, Mgrs Bad, Comm Poor with “users”, Prediction Bad]

8pt = Mgrs Bad & Comm Poor

4pt = <list of 4 causes>

3pt = morale, fail to gel group, and comm

2pt = group does not collaborate as a team

2pt = <list of Jackman's issues>

--