SE463
Fall 2025 — Final Exam

12 December 2025, 9:00am–11:30am
Instructor: Daniel M. Berry
Time allowed: 2.5 hours = 150 minutes
No aids allowed (*i.e.,* closed book).
Answer all of the questions on this exam paper.
There are 12 questions for a total of 150 marks.
Plan your time wisely: 1 minute per mark

Your Name and Student Number

In the immortal words of the yet to be born Jean-Luc Picard of Earth,

Engage!

Q1		scaled to	20
Q2		scaled to	8
Q3		scaled to	20
Q4		scaled to	9
Q5		scaled to	20
Q6		scaled to	10
Q7		scaled to	8
Q8		scaled to	8
Q9		scaled to	8
Q10		scaled to	12
Q11		scaled to	12
Q12		scaled to	15
TOTAL		scaled to	150

In this exam, if you are asked a question, which ends with a "?", you are to answer that question in the space following the current line.

In this exam, if you are asked for a simple answer, you need not justify it, unless you are also asked explicitly "Why?". However, you may always write down assumptions that can help us give you partial credit.

If an exam question directs you to list, describe, write, explain, draw, mark, put, change, modify, or anything similar, just do so.

In the exam questions,
"CBS" means "computer-based system".
"SW" means "software".
"NFR" means "non-functional requirement", a.k.a. "quality attribute".
"RE" means "requirements engineering".
"SRS" means "software requirements specification, written according to some standard, e.g., IEEE".
"UM" means "user's manual".

Note the difference between a serif font like that used for *this* clause
and a sans serif font like that used for *this* clause.
A serif font is used for ordinary text in a question, and a sans serif font is used for text in a software or requirements specification artifact.

Original Turnstile System

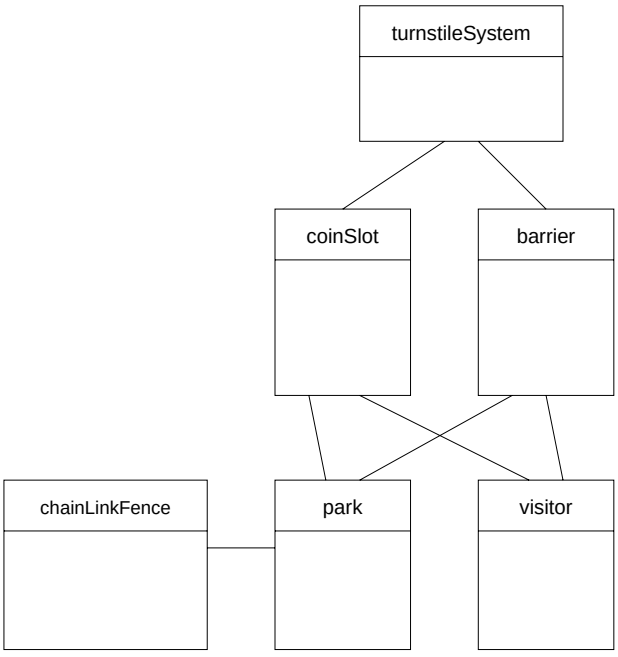
The original basic requirements given for the Waterloo Park Turnstile System are

> "Collect $1 fee from each human park user on entry to park.
>
> - Ensure that no one may enter park without paying.
> - Ensure that anyone who has paid may enter the park."

We decided to use Solution Number 2:

- Use barriers with automated coin collection.
- Use chain link fences for security.
  - There is a barrier (turnstile) through which to enter a park.
  - A person inserts a coin, the barrier unlocks, allowing the person to push the barrier and enter the park.

After some deliberation we came to the class model given below:



In this diagram, each class is given as a rectangle divided into two parts. The top part contains the class's name and the bottom part is to optionally list procedures, some of which are use cases, that the class offers to be invoked by its users. In this model, none of the procedures are given. Some questions below will ask you to fill in some, but not all, of the procedures, some of which are use cases.

Enhanced Turnstile System

Let's now enhance these requirements.

There are multiple parks in Waterloo, all controlled by a city-wide, central turnstileSystem.

Each park has one or more barriers, each

1. with its own coinSlot and
2. sitting at a gap in the park's chainLinkFence.

Each barrier has also its own QRreader that can be shown a QR code that means that the shower[1] has paid the $1.00 entry fee and must be treated exactly like anyone who has inserted a $1.00 coin, a.k.a., a loonie, into the barrier's coinSlot and is then allowed to push through the barrier to enter the barrier's park.

Thus, a QR code is usable to gain entry only at one barrier in only one park.

There is a Web site hosting parkApp, an app that allows its user to choose a park, to choose a barrier in the chosen park, to pay the $1.00 fee by any supported method, and to receive a QR code by any supported method. The received QR can be shown to the QRreader of the chosen barrier at the chosen park to gain the shower's entrance to the chosen park.

Each of Questions 1 through 6 deals with the original or the enhanced Turnstile System.

[1] show-er, one who shows, and not a shower in a bathroom :-)

1. **[20 total marks]** Domain or Class Modeling

   The diagram on the next page, all by itself, is an incomplete class model of the enhanced system. The completed part is the model given above for the original system. The unfilled boxes and other information will be filled in as a result of your answering the questions given below.

   The enhanced system has two kinds of actor classes,

   visitor and
   e-Visitor.

   The enhanced system has four interface classes,

   coinSlot,
   barrier,
   QRreader, and
   parkApp.

   (a) Fill in the top parts of unfilled boxes with actor and interface class names that are not already in the diagram. Hint: The two actor classes should be adjacent to each other in the diagram.

   (b) Provide all needed additional links between classes.

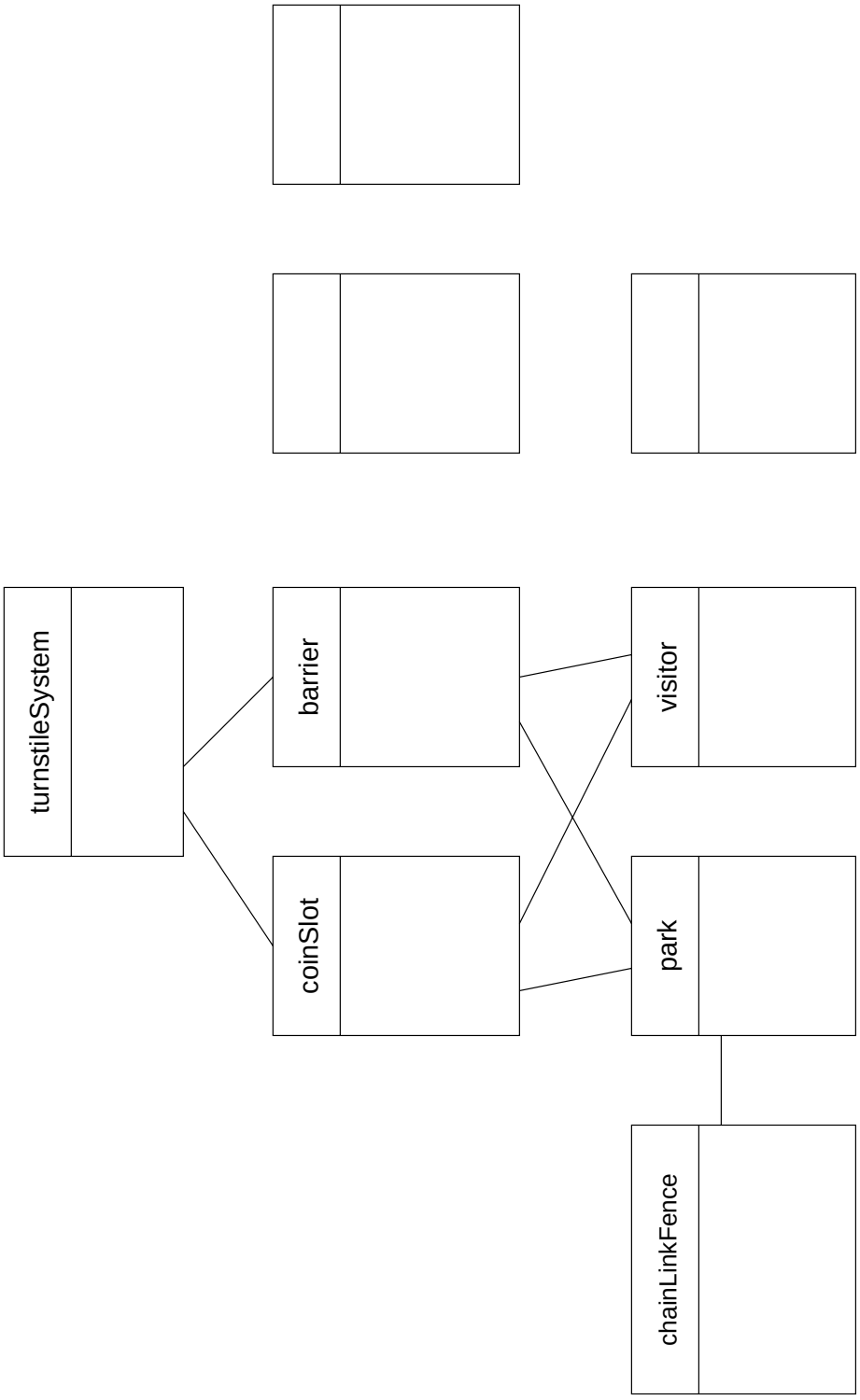
   (c) Provide multiplicities to only the classes, and not to the links.
   Use n for the multiplicity of barrier and any class with the same multiplicity.
   Use m for the multiplicity of park and any class with the same multiplicity.
   Use * for the multiplicity of any class whose multiplicity is independent of n and m.

   (d) Write here a mathematical formula expressing the necessary relation between the values of m and n:

   m _____ n

   (e) Add the stereotype ≪actor≫ to the boxes that should have them.

   (f) Superimpose a world diagram on this class model, using any notation that allows knowing for each class the regions, Sys, Env, or both, in which it sits.


turnstileSystem
barrier
coinSlot
visitor
park
chainLinkFence

2. **[8 total marks]** Use Cases in Domain or Class Model

   The six use cases (UCs) for the Turnstile System are:

   insertCoin (IC),
   pushBarrier (PB),
   showQRcode (SQRC),
   choosePark (CP),
   chooseBarrier (CB), and
   payFee (PF).

   Following each UC's name is an abbreviation for it which may be used in an answer whenever the space allowed for writing the UC's name is too small for the full name.

   It is required furthermore that

   (1) choosePark (CP),
   (2) chooseBarrier (CB), and
   (3) payFee (PF)

   be done in that order.

   In the domain model that you have been filling in for Question 1, write each UC's name or abbreviation into the bottom part, the procedure part, of the class to which it belongs.

3. **[20 total marks]** More Domain or Class Modeling

(a) In the original Turnstile System, after a visitor does insertCoin to coinSlot, coinSlot informs _____ that the entrance fee has been paid. In turn, _____ informs _____ to unlock the barrier.

A new topic for the remaining questions:

In the enhanced Turnstile System, there are procedures of the interface classes whose execution is *observable* by actors, but they are not invoked by the actors. Four, but not all, of these procedures are:

i. display"The barrier to your left is now unlocked; push it to enter the park" (DBNUPEP) _____

ii. display"This is the QRcode that you paid for" (DQRC) _____

iii. displayRedLightIndicatingThatIamLocked (DRL) _____

iv. displayGreenLightIndicatingThatIamUnlocked (DGL) _____

Following each procedure's name is an abbreviation for it which may be used in an answer whenever the space allowed for writing the procedure's name is too small for the full name.

(b) *In the domain model that you have been filling in for Question 1*, write the name or abbreviation of each of the four procedures into the bottom part, the procedure part, of the class to which it belongs.

(c) For each of the four procedures, write the name of the class that invokes the procedure in the underline after its name.

(d) Why are these four procedures *not* considered UCs?

In the following, a *valid QR code* is one that has been issued to anyone who has paid the fee to enter the park once.

(e) Consider the link between parkApp and turnstileSystem. What communication medium must be used to enable the parkApp to function as specified on a person's cellphone or computer from any location?

(f) Consider the link between QRreader and turnstileSystem. What communication medium must be used to ensure that the QRreader can access the set of *currently* valid QR codes, and not be constrained to using a finite, non-updata-ble set of preloaded valid QR codes?

(g) What problem is caused if the set of valid QR codes *is* a finite set?

4. **[9 total marks]** Use Case Modeling

   To remind you, the six use cases (UCs) for the Turnstile System are:

   insertCoin (IC),
   pushBarrier (PB),
   showQRcode (SQRC),
   choosePark (CP),
   chooseBarrier (CB), and
   payFee (PF).

   Following every UC's name is an abbreviation for it which may be used in an answer whenever the space allowed for writing the UC's name is too small for the full name.

   It is required furthermore that

   (1) choosePark (CP),
   (2) chooseBarrier (CB), and
   (3) payFee (PF)

   be done in that order.

   Fill in the skeletal UC Model (UCM) on the next page to provide these six UCs to the full enhanced Turnstile System.

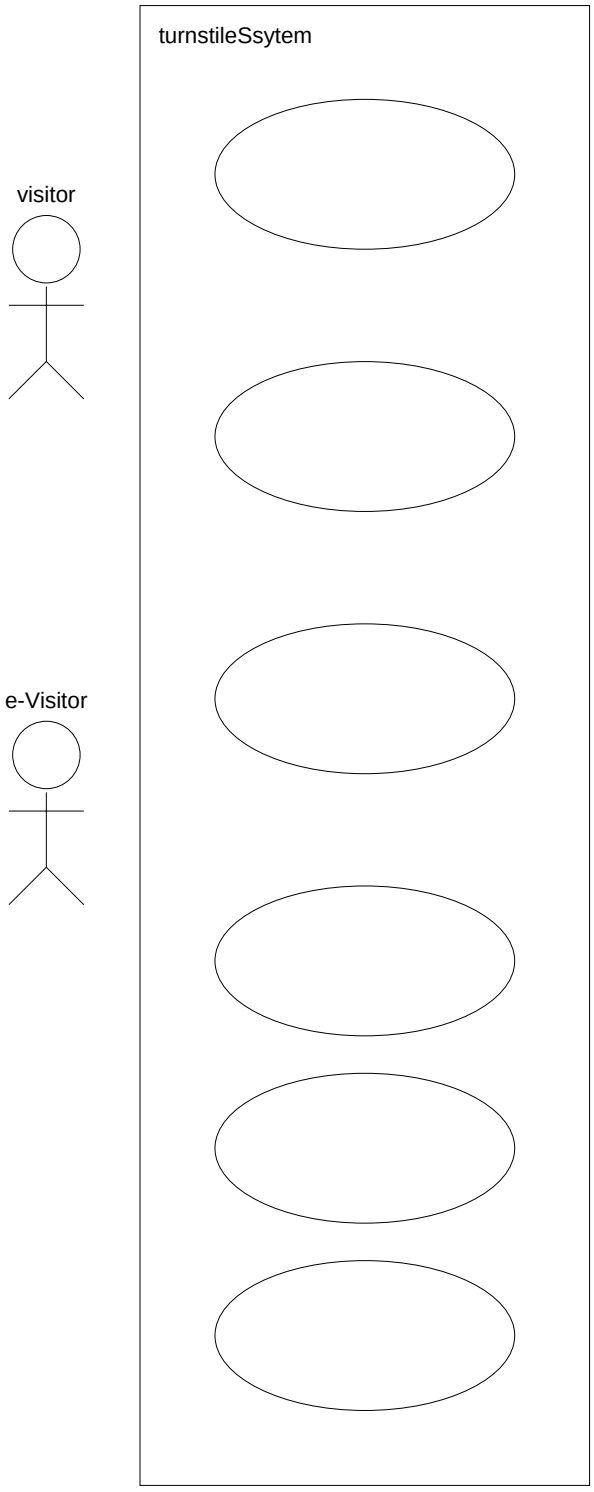
   That is,

   (a) insert every UC's name into one of the ellipses, and
   draw a link between every actor and all UCs that are accessible to the actor.
   Note that the UCM already gives the two actors. It is helpful to place UC names in the ellipses in a way that the links between actors and UC ellipses do not cross each other.

   (b) Add to this UCM an indication for every UC, in which class it resides. This indication can be

   some number of boxes, each for one class, labeled by the class's name, and surrounding all the class's UC ellipses, or

   putting in the far right of every ellipse, the name of the class in which it resides.


turnstileSsytem
visitor
e-Visitor

5. **[20 total marks]** Exceptions

In class, we provided a list of exceptions for the UCs,

insertCoin (IC)
pushBarrier (PB)

These exceptions are:

insertCoin (IC)
coin is fake
coin is not a loonie
coin is damaged
coinSlot is full

pushBarrier (PB)
barrier is locked
started push is not finished within 30 seconds

In the spaces provided below, list the exceptions of the new UCs:

showQRcode (SQRC),
choosePark (CP),
chooseBarrier (CB), and
payFee (PF)

Before you do, recall that it is required that the UCs,

(1) choosePark (CP),
(2) chooseBarrier (CB), and
(3) payFee (PF),

be done in that order.

In place of *any one exception*, it is OK to give an explanation of why the user interface you assume, which you need to describe, makes sure that the exception can never occur.

showQRcode (SQRC) Give no more than 4 exceptions.

choosePark (CP) Give no more than 2 exceptions.

chooseBarrier (CB) Give no more than 3 exceptions.

payFee (PF) Give no more than 4 exceptions.

6. **[10 total marks]** Requirements Analysis

Remember the Turnstile System's original basic requirements: (Reproduced here to avoid your having to flip pages):

"Collect $1 fee from each human park user on entry to park.

- Ensure that no one may enter park without paying.
- Ensure that anyone who has paid may enter the park.",

The current formulation of the requirements of parkApp is that a QR code is usable to gain entry *at only one barrier in only one park.*

Consider *removing* this and *only this* requirement. All others, including about QR codes remain required.

(a) Would removing this requirement prevent the enhanced system from meeting the original requirements?

(b) If your answer is "yes", describe the prevention.

(c) What is the effect of the removal of this requirement on the usability of an as yet unused, purchased QR?

(d) If this requirement is indeed removed, what UCs can be removed?

This is a new topic!
Consider allowing paying also by credit or debit card. You would need to add to the last domain model diagram, in Question 1, a class POSterminal that models a portable point of sale terminal that allows payment by credit or debit cards. DON'T TOUCH that diagram; instead, answer the questions below.

(e) The new class POSterminal must be connected by links to what classes in the last domain model diagram for Question 1?

(f) The new class POSterminal should be drawn inside which part of the last domain model diagram for Question 1, (1) Sys – Env (i.e., Sys ONLY), (2) Env – Sys (i.e., Env ONLY), or (3) Intf = (Env ∩ Sys)?

7. **[8 total marks]** Internet and E-Type Systems

Recall that what is now called the Internet started off in 1969 as the ARPAnet, which was required by Kleinrock, Cerf, et al to be a completely open network, in which anyone sitting anywhere on the network was to be able to use any other site on the network as if E were sitting at the other site. In other words, the ARPAnet was required to be open and essentially insecure.

Independently, Meir Lehman described the phenomenon of an E-type system:

- An E-type system solves a problem or implements an application in some real-world domain.
- Once installed, an E-type system becomes inextricably part of the application domain, so that it ends up altering the real world in a way that alters its own requirements.

The progression from the 1969 ARPAnet to the 2021 Internet is a classic example of this phenomenon.

(a) Describe one change to the real world resulting from the introduction of the ARPA- or Internet.

(b) Describe one new, unanticipated use for ARPA- or Internet functionality that the real world discovered as a result of the change identified in your answer to (a).

(c) Describe one requirement change to the ARPA- or Internet resulting from the change identified in your answer to (b) in the way the ARPA- or Internet was used.

8. **[8 total marks]** RE of AI (Artificial Intelligence)

   (a) Consider a task, T, that demands human intelligence for its successful performance.
   In what way is

   - deciding whether an AI for T performs T at least as well as the average human who is regarded as an expert in performing T

   the same as

   - deciding whether a CBS meets the NFR of having a fast enough response time?

   (b) Consider now a task T, such as driving a vehicle, that risks human lives.
   What must engineers do during the construction of an AI for T in order to not to be held liable if the AI ends up killing humans?

9. **[8 total marks]** Ambiguity

(a) A typical native English might write a requirement like:

A QR code is only usable to gain entry at one barrier in one park.

to say that one buys a QR code to gain entry to a specified barrier in a specified park and to no other barrier and no other park.

Rewrite the specification to mean what it is supposed to mean, while using only these same words in the same order, with the current only removed, and then adding one or more instances of only before the first word, between pairs of words, or after the last word.

_____ A(n) _____ QR code _____ is _____

usable _____ to _____ gain _____ entry _____

at _____ one _____ barrier _____ in _____

one _____ park _____ .

(b) The title of a popular TV show on Hulu is:

Only Murders in the Building

(abbreviated as OMITB). Apparently, the meaning of the title is that three amateur detectives will not investigate any crime *unless* (in its everyday meaning) the crime is a murder and it occurred in the building in which they all live. Moreover, to everyone living in this building, the building is known as "the Building".

Rewrite the title to mean what it is supposed to mean, while using only these same words in the same order, with the current Only removed, and then adding one or more instances of Only before the first word, between pairs of words, or after the last word. (Only is capitalized because it appears in a title.)

_____ Murders _____ in _____ the _____

Building _____ .

What is wrong with putting an Only after the and before Building? (Explain how the resulting meaning makes no sense!)

(c) The sentence

The sensors issue an alarm when they detect an incursion.

has *at least* four different meanings. Each meaning is generated by selecting one of three ambiguous phrases in the sentence:

1. The sensors issue
2. an alarm
3. they detect

In the space after each ambiguous phrase, give the stated number of disambiguations that adhere to the stated restrictions, if any:

i. The sensors issue, 2 meanings, each in singular form

ii. an alarm, 2 meanings, each in singular form [Hint: to what does the alarm belong?]

iii. they detect, 4 meanings, some in singular and some in plural form

10. **[12 total marks]** RE Reference Model

In the ZJVF,

$D, S \vdash R$

applied to SW development, each of D and R, being a statement about the real world, is informal, i.e., its truth is an empirical question: Is the probability that it matches the real world high enough — usually better than 95% — that humans accept it as reflecting reality? For example, the truth of anything in D or R that involves physics depends on what scientists currently know.

On the other hand, S, being a statement about the SW that is built to satisfy R, given D, *may* be formal, i.e., its truth may be a mathematical question: Can the SW be proved mathematically to correctly implement S? For example, Hoare logic can be used to prove or disprove that a C program that is claimed to sort the lines of its input into alphabetical order does indeed do so, up to the limits imposed by the hardware on which the program is run.

Conversely, there are kinds of SW for which the truth of S is an empirical question: is the probability that the SW correctly implements S high enough? If the behavior of the SW itself depends on an understanding of the real world, then the correctness of the SW is an empirical question and is thus informal.

For each of the following brief descriptions of SW, classify it as formal or informal, by writing "F" or "I", respectively in the underline preceding the beginning of the description. In the following, an LM is the learned machine, an AI that results from running a machine learning (ML) engine with data that teach the LM the correct way to do some task.

(a) _____ an LM that was taught data about actual chess games that lead to checkmate

(b) _____ an AI that exhaustively searches the tree of possible scenarios from the current chess board, using the rules of chess, to calculate a move that leads eventually to checkmate

(c) _____ an LM that recognizes X-rays that indicate a cancer

(d) _____ a program that calculates a fast Fourier transform

(e) _____ a molecular program that recognizes the tell-tale spike of a Covid virus and envelopes the virus enough to make it impossible for the virus to attack another cell

(f) _____ the TEX program

(g) _____ an operating system's process-priority-based process scheduler

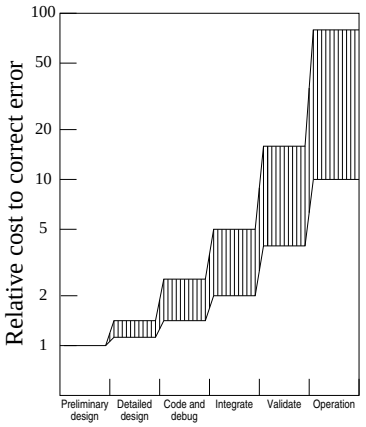
Change gears to another topic!

(h) A simple way to help ensure that a requirements specification (RS) specifies only What and not How is to restrict the RS's vocabulary to that of

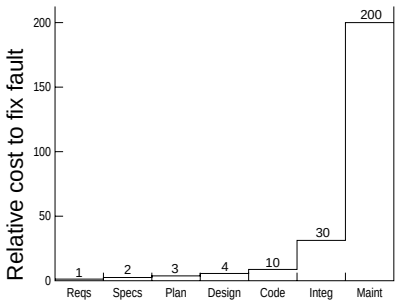
_____ in the domain model with superimposed world diagram.

11. **[12 total marks]** Cost Estimation

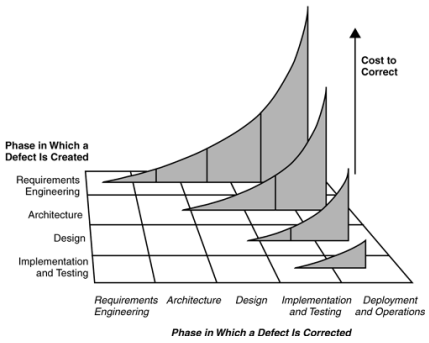
(a) The graphs and table given below are famous for explaining how much more correcting code costs than writing the code correctly from the beginning.



Phase in which error is detected



Phase in which fault is detected and fixed



Source	Phase Requirements Issue Found			
	Require-ments	Design	Code	Test
[Boehn, 1981]	1	5	10	50
[Hoffman, 2001]	1	3	5	37
[Cigital, 2003]	1	3	7	51
[Rothman, 2000]		5	33	75
[Rothman, 2000] Case B			10	40
[Rothman, 2000] Case C			10	40
[Rothman, 2002]	1	20	45	250
[Pavlina, 2003]	1	10	100	1000
[McGibbon, 2003]		5		50
Mean	**1**	**7.3**	**25.6**	**177**
Median	**1**	**5**	**10**	**50.5**
[NASA, 2010 System Cost Factors]				
Method 1	1	8	16	21
Method 2	1	3–4	13–16	61–78
Method 3	1	4	7	157–186
[Langenfeld, 2016]	1	1.6	4.9	6.7
[Hamill, 2017] Mean	1		5.1	24
[Hamill, 2017] Median	1		3	27
[IBM SSI, Pressman]		1	6.5	15
[Extrapolated & Normalized IBM SSI, Pressman]	1	5	32.5	75

Normalized Cost-to-Fix Estimates

Fill in the blanks in the following sentence to make it a one-sentence summary of what the graphs are saying.

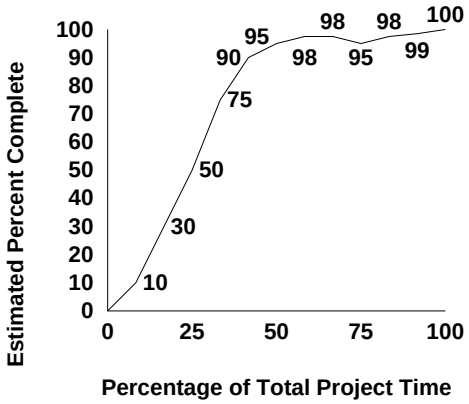
The longer one waits after the introduction of a defect to _____ the defect, the _____ fixing the defect costs.

Therefore, it pays to spend as much time as needed during requirements analysis to track down, identify, and fix missing, incorrect, and extraneous requirements.

(b) Fred Brooks observed: Every body thinks *program* when he or she should be thinking *software system product*.

- program—what you write for yourself (and thus what you know)
- system—program that interfaces with other programs, directly or indirectly, costs 3 times as much as central program (more stuff to write)
- product—program written for others, that must therefore be robust, costs 3 times as much as central program
- software system product—program that is system *and* product, costs 9 times as much as central program

Explain how Brooks's observations explain the famous graph about how inaccurate estimates of progress during a project are.



(c) A famous observation about the distribution of code in a CBS is that

- 10–20% of the code = central approximation.
- 80–90% of the code = exceptional details.

In terms of Brook's observations what is the central approximation?

In terms of Brook's observations what are the exceptional details?

12. **[15 total marks]** Linear Temporal Logic to State Machine

Remember,
$\Box$ means "henceforth", $\Diamond$ means "eventually",
$\bigcirc$ means "in the next state",
$\mathcal{U}$ means "until", and $\mathcal{W}$ means "unless".

(a) For each of the linear temporal logic formulae below, in the underscore before the formula, write "T" if the formula is true and write "F" if the formula is false. Each incorrect answer is worth the negative of one half the marks that a correct answer is worth; i.e., it's not worth guessing. Nevertheless, you cannot get less than zero in total for the temporal logic formulae.

_____ 1. $(\Box A) \Rightarrow (\bigcirc A)$

_____ 2. $(\bigcirc A) \Rightarrow (\Box A)$

_____ 3. $(\Diamond A) \Rightarrow (\bigcirc A)$

_____ 4. $(\bigcirc A) \Rightarrow (\Diamond A)$

_____ 5. $(\Diamond A) \Rightarrow (\Box A)$

_____ 6. $(\Box A) \Rightarrow (\Diamond A)$

_____ 7. $(\Box\Diamond A) \Rightarrow (\Diamond\Box A)$

_____ 8. $(\Diamond\Box A) \Rightarrow (\Box\Diamond A)$

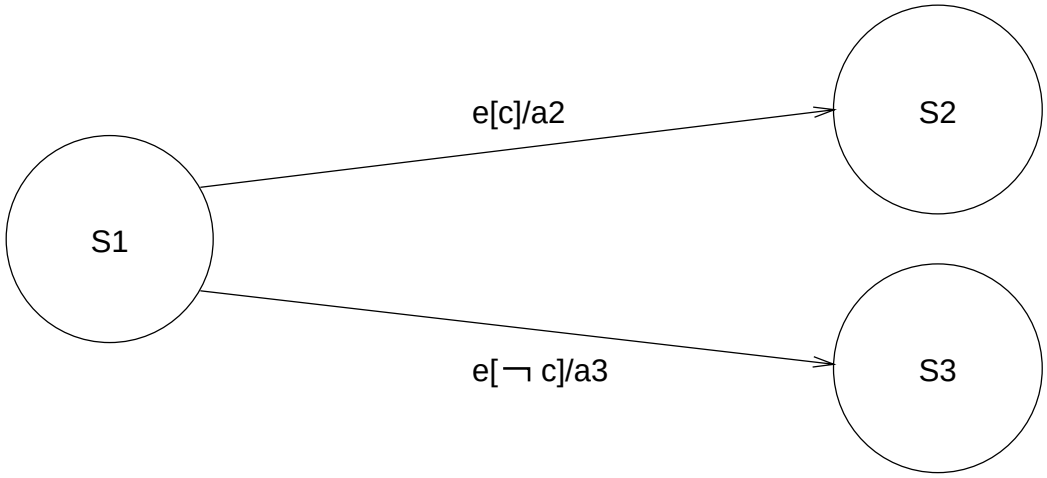
_____ 9. $(A \,\mathcal{U}\, B) \Rightarrow (\Diamond B)$

_____ 10. $(\Diamond B) \Rightarrow (A \,\mathcal{U}\, B)$

_____ 11. $(\Diamond A) \Rightarrow ((A) \vee (\bigcirc\Diamond A))$

_____ 12. $((A) \vee (\bigcirc\Diamond A)) \Rightarrow (\Diamond A)$

(b) Express the state machine transitions



in Linear Temporal Logic. In doing so, assume:

1. There are no other events that can leave $S1$.

2. The effect of an action, $a2$ or $a3$, in any computational state is not observable until the next computational state, when the state machine is in its next state, $S2$ or $S3$. You may use a state-machine action itself as the LTL predicate describing the action's effect.