1

Introduction

[There are no right or wrong answers to the first four exercises, which are intended to give you some feedback on the background of the students and to get them thinking about the value of using a software design methodology.]

1.1 The amount of time spent on analysis, design, coding, and testing/debugging/fixing depends on the methodology used. Using an OO approach, we find that our effort is split approximately 20% on analysis, 30% on design, 40% on coding, and 10% on testing/debugging/fixing problems. The exact split depends on factors such as the type of system and the amount of experience with similar systems. We have found that paying extra attention to analysis and design cuts the total development time. It is a lot easier to avoid a problem during analysis and design than it is to find and fix it later on.

One of the most difficult areas of project management is estimating how much effort a project will require. One method that we use is to break the total effort down into several tasks. We think about each task separately and estimate how much effort is required, based on our experience with similar tasks.

One major problem that we have encountered is underestimating the time and effort required to complete a project. Most software projects are finished late and over budget. We try to deal with this by carefully considering our estimates and explicitly allowing for contingencies.

Premature implementation is another problem. Because of an anxiety to complete a project, developers sometimes substitute implementation for design, resulting in systems that are hard to debug. The resulting software suffers from unwarranted assumptions and fuzzy thinking. With such an approach, developers bog down in details and find it difficult to see flaws.

In one of our projects we quickly prototyped a trivial master-slave communications system. We did the initial coding in a week without benefit of a thoughtful design. The resulting system crashed due to occasional data communications errors. We applied many patches to the system over a period of three months. Each time we thought we un-

derstood the problem only to find that we really did not. The problem seduced us into investigating many dead ends. We finally scrapped our initial design and started over again. We put more thinking into our second design and successfully completed the communications system.

Another problem we often face is uncertain or changing requirements. For example, another one of our projects changed its hardware platform several times during the course of the project, from a PC to a workstation and then back to a PC. Each time, we ported an uncompleted system, wasting time and effort. This contributed to the failure of the project. What we should have done was complete the system on one platform first and port it to other platforms later.

1.2 [Expect a wide range of answers to this question. Here is an answer based on one of our early applications at GE R&D.]

We created an editor for power system circuit diagrams that served as a graphical front end for capturing parameters needed for simulating their performance. Our main obstacle was integrating several subsystems over which we had no control. We used a structured analysis and design methodology that someone else selected over our objections. We would have preferred a rapid prototyping approach. The focal point of the system was a database, which we designed using OO techniques. Although the project failed, we salvaged several good ideas for other projects.

1.3 [Expect a variety of answers.] Many software systems suffer from these problems, including the system described in the previous answer, which was behind schedule and over budget. Contributing factors were pressures to underestimate the effort and the decision to build the system on top of several subsystems that were being developed separately. Development of one of the subsystems, a graphical interface, ran into problems of its own, and did not become available until after the target completion date of our system.

1.4 [The point of this exercise is for the students to realize that it is easy to create systems that annoy users, and to motivate them to consider the user's position.]

Software systems that truncate names is one of our pet peeves. Truncation can cause many unexpected problems. One check issuing system truncates first names to 5 letters and last names to 7 letters, resulting in checks that some banks refuse to cash. This could have been avoided by a better design.

Another annoying situation is the handling of foreign currency by vending machines. Change dispensed by bill changing machines may contain foreign currency that is refused by adjacent food vending machines.

1.5 a. Addresses can be used to identify mail recipients. The format of an address, which varies with country, often includes name, street, city, state/province, postal code, and country. An address both identifies the recipient of mail and encodes instructions for its delivery.

b. Criminal investigations can use combinations of photographs, fingerprinting, blood-typing, DNA analysis, and dental records to identify people, living and/or deceased, who are involved in, or the subject of, a criminal investigation.

c. Banks can use a variety of schemes to identify safe deposit customers. Usually name plus some other piece of information such as an account number, driver's license, or address is used. Other answers are possible.

d. Telephone numbers are adequate for identifying almost any telephone in the world. In general a telephone number consists of a country code plus a province, city, or area code, plus a local number plus an optional extension number. Businesses may have their own telephone systems with other conventions. Depending on the relative location of the telephone that you are calling, parts of the number may be implied and can be left out, but extra access digits may be required to call outside the local region.

In North America most local calls require 7 digits. Long distance calls in North America use an access digit (0 or 1) + area code (3 digits) + local number (7 digits). Dialing Paris requires an access code (011) + country code (33) + city code (1) + local number (8 digits). The access code is not part of the identifier.

e. Accounts can be used by telephone companies for billing purposes. A single account may be for one or for several telephone numbers. Account information includes account ID, name, and address. The account ID identifies the account. One of the telephone numbers in the account could be used in the construction of the ID. A bill for the service provided to all of the telephone numbers in an account can be sent to the account address.

f. There are logical as well as physical electronic mail addresses used in electronic networks. The formats depend on the particular network. A physical address is a sequence of bytes assigned to a hardware device such as a workstation or a computer at the time of its manufacture, and uniquely identifies the device. A logical address identifies a user on a system. On the Internet there are domain names (such as *edu*, *org*, and *com*). Organization names are unique within a domain and users within an organization. Thus some email addresses are blaha@computer.org and rumbaugh@us.ibm.com.

g. One way that employees are given restricted, after-hours access to a company is through the use of a special, electronically-readable card. Of course, if an employee loses a card and does not report it, someone who finds it could use it for unauthorized entry. Other approaches include a picture ID which requires inspection by a guard, fingerprint readers, and voice recognition.

1.6 [Expect a wide variety of answers. The point of the exercise is for the student to begin to think in terms of classes.]

a. Classes that you would expect in a program for newspaper layout include *Page*, *Column*, *Line*, *Headline*, and *Paragraph.*

b. Classes that you would expect in a program to compute and store bowling scores include *Bowler*, *Frame*, *Pin*, *Score*, and *BallWithinFrame.*

c. Some classes for voice mail include *Telephone, Greeting, Message,* and *Distribution List.*

d. Classes for a controller for a video cassette recorder include *Timer, Channel, Tapedeck,* and *TV.*

e. For a catalog store order entry system, classes include *Customer, Order, Store,* and *Item.*

1.7 For a variable length array the operations behave as follows:

- *Append* adds an object to the end of an array. Duplicates are allowed.
- *Copy* makes a copy of an array. All values in the array elements are copied but the objects referenced are not copied recursively.
- *Count* returns the number of elements in an array.
- *Delete* removes an element from an array. The position of the element to be deleted must be specified. All higher numbered elements are shifted down by one position. If the position is out of range, the operation is ignored. The operation returns the change in size of the array, -1 if deletion occurs, otherwise 0.
- *Index* retrieves an object from an array at a given position. NULL is returned if the index is out of range.
- *Intersect* is not defined for arrays.
- *Insert* places an object into an array at a given position. All elements at the given index or higher are shifted up by one position. An error message is printed if the position is out of range.
- *Update* places an object into an array at a given position, overwriting whatever is there. If the position is out of range, the array is extended with NULLs. The operation returns the change in the size of the array.

For a symbol table (also known as a dictionary) the operations behave as follows:

- *Append* makes sense only for sorted tables, in which case an entry goes at the end unless the table already contains the keyword. In that case the new entry replaces the old entry. Duplicates are not allowed. The operation returns the change in the size of the table.
- *Copy* makes a copy of a table.
- *Count* returns the number of entries in a table.
- *Delete* removes an entry from a table. The entry to be deleted is specified by key word. If the entry does not exist the operation is ignored. The operation returns the change in the size of the table.
- *Index* retrieves an entry from a table that matches a given keyword.
- *Intersect* is not defined for symbol tables.
- *Insert i*s not defined for symbol tables. Use *update* instead.

- *Update* adds an entry to a table. If the keyword is not yet in the table, a new entry is made. If the keyword is already in the table, the entry in the table is updated.The operation returns the change in the size of the table.

Operations on sets behave as follows:

- *Append* is not defined for a set, since elements of a set are not ordered.
- *Copy* makes a copy of a set.
- *Count* returns the number of elements in a set.
- *Delete* removes a given element of a set. If the element does not exist the operation is ignored.The operation returns the change in the size of the set.
- *Index* is not defined for a set.
- *Intersect* performs set intersection of two given sets, creating a new set.
- *Insert i*s not defined for sets. Use *update* instead.
- *Update* adds an element to a set. If the element is not yet in the set, it is added to the set. If it is already in the set, the operation is ignored.The operation returns the change in the size of the set.

1.8 [Adding more classes to each list is optional. We have given a few classes to get the student to think abstractly. We made no attempt to give exhaustive lists in the exercise.]

a. Electron microscopes, eyeglasses, telescopes, bomb sights, and binoculars are all devices that enhance vision in some way. With the exception of the scanning electron microscope, all these devices work by reflecting or refracting light. Eyeglasses and binoculars are designed for use with two eyes; the rest of the objects on the list are designed for use with one eye. Telescopes, bomb sights, and binoculars are used to view things far away. A microscope is used to magnify something that is very small. Eyeglasses may enlarge or reduce, depending on whether the prescription is for a near-sighted or a far-sighted person. Some other classes that could be included in this list are optical microscopes, cameras, and magnifying glasses.

b. Pipes, check valves, faucets, filters, and pressure gauges are all plumbing supplies with certain temperature and pressure ratings. Compatibility with various types of fluids is also a consideration. Check valves and faucets may be used to control flow. With the exception of the pressure gauge, all of the items listed have two ends and have a pressure-flow characteristic for a given fluid. All of the items are passive. Some other classes include pumps, tanks, and connectors.

c. These objects are all means for transportation. Bicycles, cars, trucks, motorcycles, and horses are used on land. Sailboats are used on water. Airplanes and gliders are used in the air. Students may discover other common traits.

d. These are all fasteners. The terms screw and bolt have similar meanings. The term screw is used to refer to wood screws, self-tapping sheet metal screws, and bolts. The term bolt

refers to a straight threaded screw. Nails, screws, and bolts are used for carpentry. Bolts and rivets are used in the assembly of machinery.

e. These are all forms of shelter. Any of them afford protection from the rain. People normally live in houses or skyscrapers, although the rest would do in an emergency. Tents, sheds, garages, barns, houses, and skyscrapers are man made. Caves are natural. All except tents are more or less permanent structures. Garages, barns, sheds, and houses are typically made out of wood, brick, or sheet metal. Skyscrapers require special construction techniques. Sheds, garages, and barns are used to store things.