Tiffany Gohnert

David Lambertson

Janine Ortiz

# **PONDER 08 : ESTIMATE COST**

## **Integrated Graduation Planner**

Often software engineers are asked to give detailed and accurate estimations based on very terse project descriptions. You are asked to do this week. Consider the following project description:

Create a system to integrate the graduation planner with the registration system so that when you complete the graduation plan, it will register you for your classes. Since everyone has their graduation plan done early, it will enable forecasting for the departments (for example, the number of sections of a class to offer in a given semester). When your day it comes to register, you are automatically registered for the class unless you desire to change your schedule.

A detailed description of the requirements are available here: [Registration and Graduation Advisor for BYU-Idaho](https://content.byui.edu/file/1682c7bf-acb5-4616-aeae-34930a15bee6/1/Ponder/416%2008%20Ponder%20SRS.pdf). You may feel that this project description is not sufficient. This is true! You may make clarifying questions to the stakeholder (the instructor in this case). He or she may answer your question or may not! (this is a real-world possibility!). Regardless of the quality of the answer, you still need to make a high-quality estimation.

Please take the following into consideration:

* There will be ten software engineers working on the project.
* Each of these engineers is "experienced," meaning they have been working on similar projects for five years and have ten years of industry experience. The team collectively has experience with system, web technology, server technology, and databases.
* The average annual salary for each of the ten will be $100,000.00 a year.

## **Two Levels of Estimation**

You need to make a system-level estimation of the project and a component-level estimation of one of the SIs. Of course, the final number you come up with is not as important as the methodology you follow to arrive at the number.

### **System-Level Estimation**

Create a system-level estimation of the entire system. To make sure that everyone is on the same page, there will be four SIs in this project. Please note that the SRS describes potentially more than 4 SIs. In other words, each of the above can be sub-divided into smaller SIs. Generally speaking, the smaller the SI the more accurate the estimate. Please do not sub-divide these big SIs into smaller ones. That is more work than necessary for this project. Our purpose is to understand the process, not come up with an accurate estimate. The four SIs are the following:

* The forecasting model
* The system integration
* The user interface for the students
* The user interface for the department chair (to view forecasted course load)

Use the methodology described in the reading. Justify all your decisions and explain where all the numbers originated. When finished, you must have an estimate for how long it will take to complete the project.

Follow the steps outlined on page 28 of the reading. A couple hints:

1. The four SIs are given.
2. See Table 4-2 or Table 4-5 for the complexity value. It could be different for each SI.
3. You will need to multiply the number of procedures described in the SRC by 2,000 (in section 6.4.2.3 of the text), which we will read about next week. Don't forget to take into account the growth factor (Table 4-4).
4. The productivity factor *Ck* needs to be in hours per ESLOC. The table in 4.4.1. is in ESLOC / month where there are 166 hours per month (2,000 hours per year / 12 months).
5. Compute the *EDevelop*, estimate the *k*, then compute the *ETotal* for the project (see Secton 4.7). When finished, compute the cost is in dollars.
6. After you have followed the methodology of the reading, make any necessary modifications based on what you have learned from the class debate. Give a brief rational as to why the adjustment was necessary and why you feel the new estimation is more accurate. This part is not optional; you must suggest an alteration. For example, the text suggests that a developer can write about 131 lines of code per month. Since a month has about 166 hours, that is about one line of code per hour. Might this be a bit pessimistic? When finished, you should have a refined estimate measured in months. This is a very important part of the assignment, constituting a full 20% of the grade.
7. You can skip step 7; we will not do the staffing equations or compute *Td*.

**Answers**

1.The 4 SI include

The Forecasting Model

The System Integration

The User Interface of the Students

The User Interface for the department chair

2.Complexity Model

The Forecasting Model: [12] Forecasting will be more logically complex than other parts of this software. It will need to determine what classes will be needed for students based on their major and their grad plan.

The System Integration: [8] This section will need to have real-time processing to which will take the forecasting model and the user interfaces.

The User Interface of the Students: [15] This will be a straight forward section which will display results for users and create a simple display for the student that will pull in the forecasting model.

The User Interface of the department chair: [12] the complexity for the department chair will be a little different because they will have to use the forecasting model to plan future classes. They will also pull information from the student interface to include in this planning.

3. Size Estimate

Procedure: 20,000, Max Size of Growth: 6.17, Mean Size Growth: 1.54

4. Since this product would need to be available to all students including those who have yet to attend a semester and are preparing for their first semester, I would classify this project as an internet public project. That means that we should have the productivity factor be 414 ESLOC/PM according to the table in 4.4.1.

5. E develop would be 20,000 ESLOC /414 ESLOC/PM. This equates to 49 person-months.

Since we are integrating this with a current platform, integration would be harder than normal so I would make K 1.35. E total = 49 \* 1.35 = 66.15.

6. In the class discussions Curtis Ho mentioned how “Size isn't not nearly to be the most important effort and schedule driver. size and complexity can definitely affect software development estimate” disagreeing with the author who states “"Size is the most important effort (cost) and schedule driver in the software development estimate”. It is for that reason we would remove the Size Estimate and replace it with a effort and schedule driver. These drivers would include a detailed analysis of schedules over the time of the project and an analysis of the effort required to complete the project.

Source: [1] Software Technology Support Center Cost Analysis Group; "**Section 4 System-Level Estimating Process**: Software Development Cost Estimating Guidebook"; Oct 2010; [Online] Available:[http://www.stsc.hill.af.mil/consulting/sw\_estimation/softwareguidebook2010.pdf](https://content.byui.edu/items/8bcbc45e-012a-48e0-800d-7082dd962f15/1/?.vi=file&attachment.uuid=2c5018f3-422e-45eb-a444-67136a1c8b5e)

### **Component-Level Estimation**

Create a component-level estimation of the forecasting SI component of the integrated graduation planner project. As with the system-level estimation completed earlier, you will need to both follow the methodology described in the text and create a refined estimation based on improvements to the methodology. Again, all the numbers, processes, and improvements must be fully justified.

You may have some difficulty coming up with specific values for many of these variables. Most of these are described in more detail in later chapters of the reading. At this point in time, you do not need to master these later chapters. Please just come up with a coarse estimate for the values.

Follow the steps outlined on page 40 – 41 of the reading. A couple hints:

1. Though the reading asks us to come up with a list of the elements in each SI (corresponding to the functions listed in the SRS), you will only need to work at the SI level. Estimation at the element level will be much more accurate but take more time than we have.
2. Use the complexity number from the System-Level estimation.
3. The effective size estimation for the Component-Level will be the *Se* numbers from the System-Level estimation.
4. The development environment is encapsulated in *Cbt*, the effective technology constant. Though we will spend a full week discussion this, you can get a rough estimate from Figure 8-1 and 8-2.
5. The development effort section is 5.5 in the reading.
6. The development schedule will be your *Td* value.
7. You can skip step 7.
8. As with the System-Level estimation, you will find certain irregularities and flaws in the numbers and equations of the text. Please validate the computed values against your intuition and experience. Make any necessary corrections and justify them. This is a very important part of the assignment, constituting 20% of the grade.

**Answers**

1. List of Elements:

The Forecasting Model

Elements:

Course dependency

Number of students

List of courses

The System Integration

Elements:

Storage size

Forecasting aspects

Aspects of Interface

The User Interface of the Students

Elements:

Course Dependency

Classes completed

Classes remaining

On Track semester

The User Interface for the department chair

Elements:

Aspects of Forecasting

Aspects of Students

2. The Forecasting Model

Course dependency [28] This will not require much just storage

Number of students [28] This will just be an array of students

List of courses [28] This also be an array of data

Logic calculations for forecasting [12]

The System Integration 8

Storage [28] This will be internal array for data storage

Forecasting aspects[12] This will require some logical complexity

Aspects of Interface [8] This will include significant interface

The User Interface of the Students 15

Course Dependency [28] This will not require much just storage

Classes completed [28] This will require internal arrays.

Classes remaining[15] This will be a stand alone system

On Track semester[21] This will require low logic with internal storage

The User Interface for the department chair 12

Aspects of Forecasting[12] This includes all elements from forecasting

Aspects of Students[15] This will include elements from Student

3. Se is 20,000 ESLOC.

4. We are going to say our developers are around average is their capability. This would put the constant at around 7000.

5. Ed= 4.72[D^.4/Cte^1.2]\*Se^1.2 person months;

Ed=4.72[8^0.4/3150^1.2]\*94500^1.2

**Ed=99.64 PM**

Ed is the full-scale development effort (PM),

D is the product complexity, =8

Cte is the effective technology constant, and =3150

Se is the effective component (SI) size (ESLOC).=20,000

6. Td=X\*3(to the power of)Ed months

Td=31.936

Td = development time (months),

X = an empirically derived constant between 2.9 and 3.6 depending on software product complexity, and =3.2

Ed = development effort (PM). =99.64

7. Don’t need

8.

Ed= 4.72[10^0.4/Cte^1.2]\*Se^1.2 person months;

Ed=108.94 person months

In the class discussion Tiffany Gohnert mentioned how D isn’t something that can be measured therefore we have removed it from the equation and replaced it with a standard number. The author defines complexity as how hard the design or implementation is to understand. However one is left to ask what is the baseline to determine what understanding is. You cannot measure something that is different depending on the person. For some one thing might be easier to understand than it might be for others. In order to use complexity then the author must better define understanding.