

Software Project Management

(A.Y. 2016/2017)

Exam Solutions

February 28th, 2017

Preamble

The Dummy Software Company (DSC) has been asked to develop a complex software system for the optimization of the fleet of a delivery company. The software should optimise the daily paths of each truck, according to the packages that need to be delivered. The software will take in input an XML file containing a list of packages to be delivered within the day. For each package the file includes the size, the weight, and the delivering address. The software will produce a report in which for each truck the following information are included:

- an ordered list of packages. The list will be used by the warehousemen to load the truck.
- the truck route for the day as well as the sequence of stops needed to deliver the loaded packages
- for each stop a list of packages that need to be unloaded

The software should include a graphical user interface to visualize the route for each truck on a map, and to permit to a user to modify the proposed plans.

You, as an employee of the company, have been nominated Project Manager and the management is now asking you to provide some forecasting in order to decide on how to proceed with the project. Therefore in order to define a first estimation, and according to the summary description above, you derive a first draft of the architecture for the system, shown in Figure 1, in which the components are mainly organized in a pipeline.

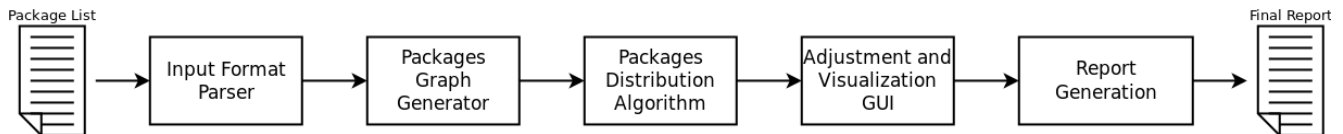


Figure 1: System Architecture

Type of Component	Difficulty											
	OE		OM		OH		NE		NM		NH	
	SD	JD	SD	JD	SD	JD	SD	JD	SD	JD	SD	JD
Control	4	21	6	27	8	30	9	33	12	40	19	49
Input/output	5	15	7	17	11	19	15	23	23	34	31	43
XML parsers	3	6	5	12	9	18	12	23	15	29	18	35
Algorithm	3	10	5	19	20	30	19	32	36	62	52	82
Data Structures Manipulation	3	8	15	30	22	33	32	45	43	54	41	67
User Interfaces	2	8	4	10	10	22	17	35	30	50	43	87
Time Critical	2	5	3	9	12	18	14	26	32	43	41	78

Table 1: Average effort for typical components (SD – Senior Developers/Analysts, JD – Junior Developers)

Additionally you can rely on data collected by your organization in past projects. In particular Table 1 reports the weeks needed in order to develop components with different characteristics and complexity. The table distinguishes the total effort spent by senior and junior employees. The gross salary per week of DSC employees are listed in the following, and it has to be noted that the company generally experiment a 60% of overhead:

- *Senior developer:/Analysts* 2500\$

- *Junior developer: 1500\$*

Successively in order to derive a more reliable estimation you sketch a set of workpackages (WPs) and tasks needed in order to complete the project. WPs and tasks are detailed in the following list (within parenthesis we report the expected duration in weeks – Ws, the number of continuously active senior developers/analysts – SD, and continuously active junior developers – JD when the task is active):

- **WP1 - Overall System Engineering**

- T1.1 - Requirements Elicitation and Analysis (Ws:10, SD:2, JD:1)
- T1.2 - Architectural Refinement (Ws:12, SD:2, JD:2)
- T1.3 - Collaborative Infrastructure Setting (Ws:3, SD:0, JD:2)
- T1.4 - Integration and System Testing Plan (Ws:6, SD:1, JD:2)
- T1.5 - Integration and System Testing Execution (Ws:10, SD:0, JD:3)

- **WP2 - Algorithm for optimal allocation of resources**

- T2.1 - Study of literature (Ws:2, SD:1, JD:1)
- T2.2 - Algorithm definition and implementation (1st iteration) (Ws:8, SD:2, JD:1)
- T2.3 - Algorithm definition and implementation (2nd iteration) (Ws:12, SD:2, JD:2)
- T2.4 - Algorithm evaluation and testing (Ws:4, SD:0, JD:2)

- **WP3 - Data Formats and Components interaction APIs**

- T3.1 - Data exchange format and API definition (Ws:4, SD:0, JD:2)
- T3.2 - API implementation (Ws:15, SD:0, JD:2)
- T3.3 - Component testing (Ws:6, SD:0, JD:2)

- **WP4 - Graphical User Interface**

- T4.1 - GUI Design (Ws:4, SD:1, JD:1)
- T4.2 - GUI implementation (Ws:10, SD:1, JD:2)

The following list reports the dependencies among the tasks listed above where “ x depends from y” is written as “ $x \leftarrow y$ ” (for the sake of space we write “ $x \leftarrow y, z$ ” as an abbreviated form of “ $x \leftarrow y$ ” and “ $x \leftarrow z$ ”).

- $T1.4 \leftarrow T1.1$
- $T1.5 \leftarrow T1.4$
- $T2.2 \leftarrow T2.1$
- $T2.3 \leftarrow T2.2$
- $T2.4 \leftarrow T2.3$
- $T3.1 \leftarrow T1.2$
- $T3.2 \leftarrow T3.1$
- $T3.3 \leftarrow T3.2$
- $T4.2 \leftarrow T4.1$
- $T1.5 \leftarrow T2.4, T3.3, T4.2$
- $T4.2 \leftarrow T4.1$

Exercise 1.

On the base of costs experimented in previous projects for the development of similar software components, derive a first estimation for the total costs of the system. Please justify your choices. **4 points**

Solution:

The exercise mainly asks to apply the Walverton strategy to provide an estimation for the cost of the software. From the architecture reported in Figure 1 it results that the system has been already structured in 5 different components organized in a pipeline. According to this we need to assess the complexity and novelty of the component and then identify the effort needed to develop it as illustrated in Figure 1. In particular let's assume the following evaluation for the 5 components:

- *Input Format Parser*: it could be reconducted to the category "XML Parsers" components. Here we judge it not particularly novel and easy to develop so the effort to consider is 3 weeks SD and 6 weeks for JD
- *Package Graph Generator*: it could be reconducted to the category "Data Structures Manipulation" components. Here we judge it a quite standard component of medium complexity. The effort to consider is 15 weeks SD and 30 weeks JD
- *Package Distribution Algorithm*: the component can be reconducted to the category "algorithm" components. It could seem that the optimization algorithm is rather new and complex so its development will require 52 weeks SD and 82 weeks for JD
- *Adjustment and Visualization GUI*: the component can be reconducted to the category "User Interfaces" components. We could judge the component as not particularly novel but the complexity given the interaction with an external map could be rather high. The effort required will be 10 weeks SD and 22 weeks JD
- *Report Generation*: this component can be reconducted to the category "Input/Output" components. It does not seem to be particularly novel nevertheless some of the constraints on the output could be effort prone and we judge it as complex. Correspondingly the effort will be 11 weeks SD and 19 weeks JD.

According to the listed estimation the development of the software will require a total effort of 91 weeks SD ($WSD = 91$) and 159 weeks JD ($WJD = 159$).

In order to derive an estimation for the total cost (TC) we need to multiply the effort for the week cost for SD (CSD) and JD (CJD). In doing this we need to consider the overhead ($OH = 0.6$). So the final results will be:

$$TC = (WSD \times CSD + WJD \times CJD) * (1 + OH)$$

Therefore the total cost estimated using the walverton strategy is equal to: **745.600\$**

Exercise 2.

Apply the CPM strategy to derive a first estimation of the duration of the project, the critical path and the admitted float for each activity. **8 points**

Solution:

To apply the CPM strategy we should derive the graph for the precedence network, and then we will proceed with a forward and a backward pass on the graph. After having performed these steps the final results will be something similar to what is represented in Figure 2.

From the figure it results that the **critical path** is constituted by the following sequence of activities:

- T1.2 \rightarrow T3.1 \rightarrow T3.2 \rightarrow T3.3 \rightarrow T1.5

Exercise 3.

Discuss the issue with an initial assignment only based on an early start strategy, and propose a possible alternative and admissible resource distribution. In doing this consider that activities cannot be split in subactivities and that it is not considered convenient to stop and restart them. Moreover it is not possible to dilute the effort on the activities so to prolong their duration within the float. Finally consider that the company charge your project with resources that are released and successively reassigned to the project for durations smaller than 3 weeks. According to this derive the total cost of the project (only considering personnel related costs). **11 points**

[illegible]

project after only one week. It can be noted on the other hand that the project has a peak in the usage of developer that could be smoothed moving the activities within their calculated floats. The objective of this optimisation is in general to reduce the maximum number of required resources. Table 3 report a possible revised plan. It can be noted that the revised plan permits to reduce the maximum number of SD from 7 to 5 and the maximum number of JD from 8 to 6. At the same time there are no resources released by and reassigned to the project for a period smaller then 3 weeks.

You are a member in a Scrum team developing a software system since a couple of months. In the middle of a sprint planning meeting the Product Owner suggests to define dependency relations among all the stories in the product backlog. In this way he intends to constraint the selection of the tasks to accomplish made by the team members during the sprint execution. In his opinion this will permit to have a much better organization of work, avoiding possible deadlocks. You intervene in the discussion suggesting that this is not really in line with Scrum practices. On which evidences would you base your argumentation?

4 points

Answer:

Question 2.

Shortly summarize the Agile values and their meaning for a development team.

3 points

Answer:

Agile methodologies are based on four simple values that as reported in the Agile Manifesto:

- Individuals and interactions over processes and tools: understanding people and their way of working is more important and effective than imposing processes and supporting tools.
- Working software over comprehensive documentation: working software is the best tool to share a common understanding with the users.
- Customer collaboration over contract negotiation: agile methodologies consider the customer as part of the team and recognize that in order to really satisfy user requests it is important to have the customer in the loop and to have means enabling and making communication easy
- Responding to changes over following a plan: again this value stresses the fact that customer satisfaction is the primary objective of an agile approach. Therefore it is important to consider and address change requests even though they emerge late in the project.

As said many times also during SPM classes, the four statements do not intend to say that the aspects related to the right part of a sentence are not important, instead the sentences want to clarify that the aspects related to the left part of a question are more important for a successful team.

Question 3.

The Agile Software Company has been asked to develop a software system. For long time now, the agile team has adopted the SCRUM methodology, and from a first analysis the team has inserted in the product backlog already 15 user stories that after the evaluation of the team, using fibonacci numbers, it summed to 123 story points. Assuming that till the end of the project 77 additional story points will be added to the backlog and that the team has a velocity of 25 story points for iteration, how long the project will last in total if each iteration lasts 30 days?

3 points

Answer:

In total the project will require to manage 200 story points. Considering a team velocity of 25 story points for sprint it immediately follows that the project will last 8 sprints and then 240 days.