Université d'Ottawa · University of Ottawa

School of Electrical Engineering and Computer Science

ELG 5100

Assignment 2 Solution

Q1.

a) BCWP: budgeted cost of work performed = 5 + 11 + 12 = 28000 $
ACWP: Actual cost of work performed = 4 + 11 + 13 = 28000 $
BCWS: budgeted cost of work scheduled = 5 + 11 + 12 + 12 = 40000 $

Cost performance index:
CPI = BCWP/ACWP = 28/28 = 1.0

Schedule performance index
SPI = BCWP/BCWS = 28/40 = 0.7

Critical ratio
CR = SPI * CPI = 0.7

i) From the CPI (cost performance index) we can say that the project is on budget since the index is one. We can also see by looking at the numbers that the project's actual costs are very close to the budgeted costs. However, we must keep in mind that D is not finished yet and so it is not taken into account in our calculations. Depending on D, we might have a problem with budget (or not, hopefully). A good project manager would reserve judgment until D is finished too.

ii) From the SPI (Schedule performance index) we can say that the project is late and doesn't seem to run smoothly on schedule since the index is less then 0.8, the project scheduling should be checked. We can actually see from the numbers in the table that for all 3 milestones the real values were off from the scheduled values: first milestone took 4 instead of 5 (shorter), the second 12 instead of 10 (longer), and the third 4 instead of 5 (shorter), and more importantly, the fourth one is not finished yet even though it was supposed to be done by month 25.

iii) As for the overall health of the project, the CR (critical ratio) shows a project that is not in very good health since it is less then 0.8. In this case the schedule is the problem.

b) If the current trend continues, the estimated time at completion is:
ETAC = Original Time / SPI = =30/0.7= 42.9 ≈ 43 months

The estimated cost at completion ECAC = BAC/CPI,
where BAC = budget at completion = 5+11+12+12+11= $51K
ECAC = 51/1.0 = 51000 $

Q2.

a) Modularity = Cumulative number of broken SLOC due to N defects,
where N = C0+C1+C2.
Modularity = B/N = (500+5000+300)/(20+100+20) = 5800/140 = 41.43 SLOC/defect

b) Adaptability = Cumulative effort spent fixing N / N
= E/N = 5+30+15/140= 50/140 = 0.36 man-days/defect

c) Maturity = Usage time / Number of defects C0 C1 = UT/(C0+C1) = 720/(20+100)
= 6 hours/defect

d) Two ways to solve this problem:
1- Maintainability = scrap ratio/rework ratio
Scrap ratio = $B/SLOC_T$ = Breakage/total SLOC = (500+5000+300)/15000 = 0.387
Rework ratio = Rework effort/Development_Effort = E/Dev_effort
= (5+30+15)/300 = 50/300 = 0.167
=> Maintainability = 0.387/0.167 = 2.32

2- Maintainability = ratio of productivity of maintenance to productivity of development
Productivity of maintenance = Breakage/ Rework effort = (500+5000+300) / (5+30+15)
= 5800 / 50 = 116 LOC per man-day
Productivity of development = $SLOC_T$ / Development_Effort = 15000/300
= 50 LOC per man-day
=> Maintainability = 116/50 = 2.32

Q3.

a) Two ways to solve this problem:

1- Project velocity is the total of estimates of the user stories done during an iteration (official definition); hence, Project Velocity = 3 + 7 + 6 + 5 + 10 + 10 + 8 = **49 man-days/iteration**

2- Total number of developers = 7 persons
Total original estimated days = 3+7+6+5+10+10+8 = 49 days
Total actual days needed = 5+10+10+8+10+7+9 = 59 days

"a day's worth of estimated work" takes on average 59/49 = 1.20 actual days
each person works, on the average, 59/7 = 8.43 days/iteration (not everyone is fulltime!)

hence, Project Velocity = 7 * 8.43 / 1.20 = **49 man-days/iteration**

b) In the next iteration, we will have a velocity of 49 man-days per iteration, which means we can assume that we will deliver 49 man-days worth of originally-estimated work in the iteration, if the iteration was 10 working days (2 weeks). However, the iteration is 9 days, due to the long weekend; therefore we can expect to deliver:

$(49/10)\times 9 =$ **44.1** days or original estimates.

We have a total of 51 man-days in the estimates table, leaving $51 - 44.1 = 6.9 \approx 7$ days of work that will not fit in iteration 2. We can move D2 and D3, for example, to iteration 3. Other tasks will make it to iteration 2.

About A2: note that A2 needs 10 days to be done which, compared to the 9 days available in the next iteration, seems not doable in the next iteration. However, the table doesn't say how A2's 10 days are scheduled. It could be that more than one developer do it in parallel, in which case it will fit in the next iteration. But if A2 needs 10 days sequentially, then of course it will not fit in the next iteration. But, since the problem is explicitly saying "Ignore actual assignment of the tasks to specific developers", either of the answers will be acceptable.

Q4.

a)

Methodology	**Team size**	**Iteration length**
Scrum	5 to 9 (so 9 is max)	1 week to 1 month (so 1 month is max)
Extreme Programming	6 to 12 (so 12 is max)	1 to 3 weeks (so 3 weeks is max)
Crystal Method	up to 80, depending on class of project	1 to 4 months (so 4 months is max)

b) Yes, we can. We can divide the team into 6 sub-teams of 8 or 9 people and do Scrum for each sub-team separately.

Q5. Nailing down requirements for a software project at its beginning is like trying to hit a moving target. But the idea is to at least aim towards the target with the least deviation that we can at that point in time. With each iteration, the target becomes clearer and our aim improves too. If we have a good start, as opposed to an ad-hoc start, we will have to make fewer adjustments (i.e., less budget, less resources, …) on the way to reaching our goal. Even if the requirements change drastically, the work on requirements gathering, WBS, estimation, planning will help us make risks evident early. By requirement change management, duties are clear to customers and project team, which could prevent conflicts and disputations in the future.