Evan Louie
72210099

Question 1

a) **BCWS (Budgeted Cost of Work Scheduled)** – Approved/allocated budget to complete scheduled task within a given time.

ACWP (Actual Cost of Work Performed) – Actual cost which has been spent (instead of the budgeted mentioned in BCWS).

BCWP (Budged Cost of Work Performed) – The budged cost of work which has actually been performed/completed within a given time.

End of First Hour:
Budged Cost = $1/5^{th}$ of \$50.00 = \$10.00
Scheduled Completion= $1/5^{th}$ of total time = 20% scheduled completion
Actual Completion = 150 Cookies = 75% of Scheduled Completion = 15%
Actual Cost = \$9.00

BCWS = Budgeted Cost * Scheduled Completion Percentage
BCWS = \$50.00 * 20%
BCWS = \$10.00

BCWP = Budged Cost * Actual Completion Percentage
BCWP = \$50 * 15%
BCWP = \$7.50

ACWP = Actual cost
ACWP = \$9.00

b) Behind schedule by 50 Cookies (15%) and over budget \$1.50 for current work completed.
SV = BCWP – BCWS
SV = \$7.50 - \$10.00
SV = -\$2.50

CV = BCWP – ACWP
CV = \$7.50 - \$9.00
CV = -\$1.50

SPI = BCWP/BCWS
SPI = 7.50/10.00
SPI = 0.750

CPI = BCWP/ACWP
CPI = 7.50/9.00
CPI = 0.833333

Summary: The -$2.50 Schedule variance indicates we are behind schedule and are short $2.50 in product. The SPI value of 0.750 indicates the project team is less efficient the time allocated to the project and has only completed 75% of the planned completed. The -$1.50 Cost Variance indicates the cookie manager is currently $1.50 over budget and the CPI value of 0.833 indicates we are currently only making 0.833 cookies per the cost of what should be a whole cookie (we are running a deficit for every cookie made).

c) EAC = ACWP + ((BAC-BCWP)/CPI)
EAC = 9.00 + ((50.00-7.50)/0.8333)
EAC = $60.00

SAC = 5 Hours/0.75
SAC = 6.667 Hours

VAC = $50.00 - $60.00
VAC = -$10.00

d) The earned value approach doesn't apply well to software due to the modular/component based nature of software systems. Said components can vary largely in difficulty/cost and are hard to correctly assign a value to. Other reasons can be that software is only a component of a larger project, existence of numerous subcontractors, which cannot be audited for performance. However the largest difficulty is that Earned Value is dependent on having a work breakdown structure, which is very difficult in software development.

e) User stories (user-story-points) and burn-down/up charts are the newer means of measuring/assessing progress of software projects. They are more suitable than earned value due the fact that they measure progress in feature/use-case completion; which are much more measurable than 'value'

Question 2

a) A Gantt chart is a bar chart which reflects the start/finish dates for a group of tasks. They will also reflect task dependencies and can be used to implement the critical path method to discover bottleneck tasks.

In terms of what we have been discussing in class; the Gantt chart estimates the **time** required by **people** to complete an assigned (**intent**) amount of **work,** which goes towards the completion of a **product**. This project is broken up into smaller tasks each of which should be assigned an optimistic, normal, and pessimistic time requirement that are all used to account for **uncertainty** of said task ([optimistic + 4*normal + pessimistic]/6). **Cost**, although not directly measured in the Gantt chart, can be best measured via calculation with the estimated time required to complete the project (which is a Gantt chart makes available). You can measure the **value** of varying tasks by looking at which tasks act as a dependency to other tasks; some tasks

may act as a dependency for many/all other tasks, as such hold a very high value. **Risk** is reflected via the showing of dependencies of tasks; a task which acts as a dependency for many other tasks will have an intrinsically higher risk simply due to the fact that the other tasks can't proceed without it's completion. Although **quality** is not intrinsically measured within a Gantt chart, you can add quality-assurance task that can be used as a dependency to proceed onto further development; this will allow you to incorporate QA into a prospective timeline.

b) Gantt charts do not work very well with the older ways of project management (waterfall/sequential-time-gate) styles; as it was very difficult to do proper time estimates based on one large planning/requirements-gathering stage. However, with the migration towards more agile/iterative development practices, it has become much more reasonable to acquire more accurate time/cost estimates (albeit, it might require a few iterations to refine said estimates to a usable state). So in an ironic twist, with the rise of new methodologies to run a project such as agile, an older style of estimation such as a Gantt chart actually becomes much more readily usable within software development.