Worksheet 4: Agile Project Management

Updated: 3rd April, 2020

Minor revisions: 23rd March, 2022

1. Burn-Up Charts

Obtain a copy of `ProjectData_Incomplete.ods` (or the equivalent `.xlsx` file) from Blackboard. Open it in either LibreOffice Calc or Excel. This worksheet shows project data for a software tool to visualize space exploration with space crafts.

(a) Your task is to fill in the missing cells, with background colours 🟦, 🟧 and 🟨.

Note1: Although you can calculate some of these values manually, it's good (andultimately easier) to become familiar with spreadsheet formulae.

Here's a crash course in spreadsheet formulae (in case you're unfamiliar with them).

- They're like a mini programming language. A cell containing a formula shows an automatically calculated value.
- A formula always begins with "=". If you miss this out, the cell is just a fixed value, not a formula.
- Each formula is a single expression, which can include numbers, strings, arithmetic, cell references like "D4" (which work like variables), cell ranges like "D4:D10" (which work like arrays/lists of values), and functions.
- Functions include `SUM()`, `SUMIF()`, etc. There is a wealth of documentation on each one. They return a value, and most of them take parameters, just like in Python or Java. These parameters are often cell ranges (like arrays/lists).
- Copying and pasting formula cells *alters the cell references (and ranges) they contain*, keeping them relative to the cell containing the formula. For instance, if cell B1 contains a formula referring to A1, and you copy cell B1 to G15, then G15's formula will refer to F15.

  Often this is the desired behaviour, but sometimes it isn't. You can prevent it by using "`$`"; e.g., "`$A$1`" is an absolute reference that will remain as-is when you copy and paste a formula that contains it.

(i) Fill the correct cell for "sum of task durations".

(ii) Find the correct cell for "planned progress per day".

Note2: The "planned project duration" comes, in principle, from our AOA or AON graph, after we've calculated the earliest finish times for all the tasks. In this scenario, just take the value as given.

(iii) Fill the “planned progress” column, from 0–80 days. (We plan to finish the project atday 79.)

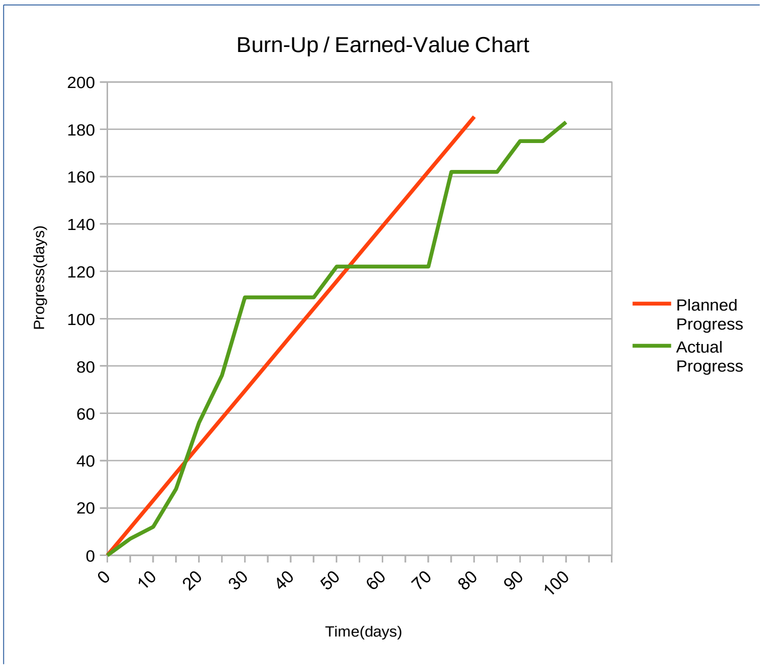
(iv) Fill the “actual progress” column, from 0–100 days.

Note3: To do this, use a formula based on the spreadsheet “SUMIF” function.SUMIF takes three parameters:

- First, a range of cells whose values we want to check.
- Second, a string that represents an expression to check for each cell in that range. Often this is of the form ”<” & D4 ,”>” & A6 , etc. (Where “&” joins strings together, like “+” in Java and Python.)
- Third, a range of cells whose values we want to add up. This should be the same size as the first range.

For instance, “=SUMIF(A1:A5, ”>=” & H1, B1:B5)” will add up the values in B1 to B5, but only where the corresponding value in column A is greater than or equal to H1.

Ultimately, you should be able to reproduce this graph:



(b) This graph shows the actual progress reaching completion. What does it tell you about how the project went, in hindsight?

(c) Consider an alternate world where the project progress turned out differently. For this task , use the Worksheet4Q1(c) worksheet.

- First, we're now only up to day 40, so leave the "actual progress" column blank beyond that point.
- Second, delete the "actual finish" numbers from the spreadsheet, and instead insert the following (leaving "[unchanged]" values unchanged, and make all "[blank] values blank/empty).

  Look at table 1 below for a guidance.

Table 1 : new project scenario

Task	**Estimated Duration**	**Actual Finish**
1. Orbital Mechanics	*[unchanged]*	31
2.1. Time Tracking	*[unchanged]*	1
2.2. Camera Angle	*[unchanged]*	4
2.3. Camera Fly-Through	*[unchanged]*	4
2.4. Bookmarks	*[unchanged]*	2
3.1. Orbit Visualisation	*[unchanged]*	3
3.2. Textures	*[unchanged]*	11
3.3. Popup Labels	*[unchanged]*	9
4.1. Planet Rendering	*[unchanged]*	24
4.2. Star Rendering	*[unchanged]*	35
4.3. Asteroid Rendering	*[unchanged]*	*[blank]*
4.4. Spacecraft Rendering	*[unchanged]*	*[blank]*
5.1. Solar System Data Entry	*[unchanged]*	*[blank]*
5.2. Star/Galaxy Data Entry	*[unchanged]*	*[blank]*
5.3. Extrasolar Planet Data Entry	*[unchanged]*	*[blank]*
5.4. Spacecraft Data Entry	*[unchanged]*	*[blank]*

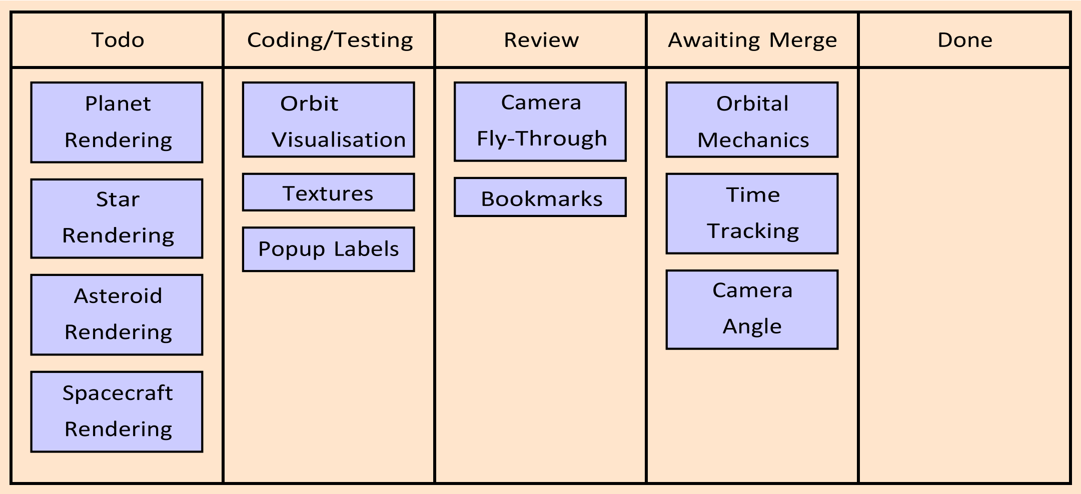
The graph should look quite different now.

What conclusions would you draw about theproject at this point?

What action would you take?

2. Kanban Boards

Consider the following kanban board situation, and say we have a WIP limit of three.



Todo	Coding/Testing	Review	Awaiting Merge	Done
Planet Rendering	Orbit Visualisation	Camera Fly-Through	Orbital Mechanics	
Star Rendering	Textures	Bookmarks	Time Tracking	
Asteroid Rendering	Popup Labels		Camera Angle	
Spacecraft Rendering				

(a) What does this mean for our (current) ability to code, review, and merge new tasks?

(b) Say our client wants "Spacecraft Rendering" to be completed as soon as possible. If we ignore potential task dependencies, but keep to the WIP limit, what would we need to do here?

(c) At a minimum (in order to adhere to WIP limit), how many other tasks will end up in the "Done" column once we're done with "Spacecraft Rendering"?

(d) Would this number reflect what is actually likely to happen? Why, or why not?

(e) Say that "Spacecraft Rendering" is finally merged, and thus in the Done column, but the client is not satisfied with the result. How would we handle this?

(f) Your team has decided that, in future projects, each task will have associated documentation written for it. For instance, for "Time Tracking" (if it were to be done again) there would be a section in the user manual describing how it works. To write this documentation properly, the code first needs to have been completed. The documentation needs to be subject to review and merging (along with the code).

(g) How would you adjust the kanban process to represent this?

(h) On a different note, say you wanted to use a kanban board for studying ISE topics. What would it look like?

3. Scrum

(a) Say you're working with a Scrum sprint length of 4 weeks, and that period has come to an end. Over the next two days:

(i) What are the key events that take place?

(ii) What are the responsibilities of the scrum master, the product owner, and the development team in these events?

(b) Where would planning poker fit into Scrum?

(c) How does Scrum handle changes to requirements, mid-way through a project? Who is immediately responsible?

4. Code review and checklists

Consider the software tool to visualize space exploration with space crafts, used in the Q1 and Q2 above.

(a) At which point of the project workflow a code review would happen?

(b) Is formal software inspection method a good choice for this project or not? Indicate reasons for your answer.

(c) Write down 5-8 code review checklist items for this project.

End of Worksheet