Milestones and Project Management Notes

Academically, there is one essential submission of the project. It is due on the day of your last class in this course. Extra time is available at 20% per day thereafter. The absolute deadline, after which no submissions will be accepted (even if you upload it), is the last day of classes in the term. Final assessment and course marks are based on that. Your professor may request additional submissions beforehand.

The milestone dates in our project plan are there to organize and coordinate the progress of everyone in the group. A project is a cooperative effort, not independent silos that hopefully (magically!) stand together.

Milestone dates in project management are not due dates in the academic sense. If you are late on an academic solo assignment, it is your problem. If you are late on an academic group project, it becomes the group's problem in getting the work done while keeping their other commitments and retaining their sanity. If you miss milestones on an industry project, it can easily become the company's problem. (The company's solution is eliminating the problem, that will be *you*.)

Whether academically or professionally, missing milestones means missing opportunities for success. Related to this is Software Development Is Almost 100% about Communication.

Milestone dates in industry are essential to meet if the project is to be delivered on time for all stakeholders. The many stakeholders include project managers, group leaders, programmers, systems and business analysists along with key users who will perform integration, system, and user acceptance testing (UAT). Those who benefit from the on-time promotion of your project into production are the customers, users, and clients (and professors) who all have their own deadlines which can only be met by the on-time deployment of a fully tested application.

There are often multiple concurrent projects that must be completed on time.
Other projects may depend on the timely delivery of your project.

A two day delay of a critical module can turn into a deployment that is two months late:

If the programmer is 1 day late delivering a module, the analyst is +2 days late verifying unit tests, the leader is +4 days late assembling modules for integration testing, and QA/QC will be +8 days late completing system testing.

Because the programmer was one day late, the schedule is now over two weeks late.

UAT will be further delayed by many days if that team agrees to work evenings and the weekend, but it is usually weeks before they can coordinate and reschedule UAT.

A late project is often a rushed project. Late projects become later due to rework when bugs and problems are found because the original work was done under pressure. At each stage, testing may reveal rework needed at the previous stage. Each stage of rework can double the lateness. If the project is two weeks late, rework means the project will be at least four weeks late, but only if everyone retains their (wildly delusional) optimism and needs no sleep.

After another round of integration, system, and UAT, Go Live will only be two months late if you are lucky, or the application will be installed with insufficient development and outstanding issues. Nobody wants their system to be the next Canadian Phoenix Pay system.

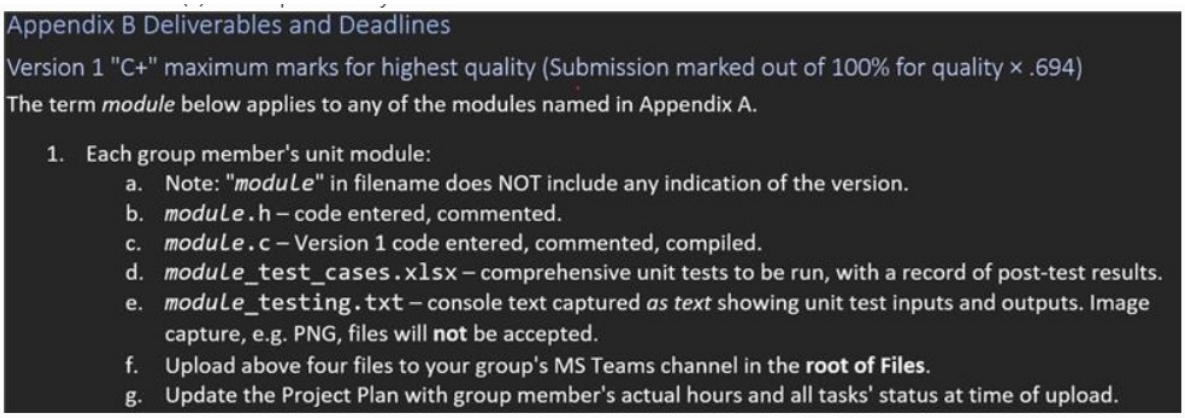
If it could not be on time the first time, when will there be time to fix it later?

[Wednesday 23:39] CPR101 group project leader posted [professors did not make this up]

Notes for next time

Hey team here are a couple of notes for the next versions as getting v1 done was significantly more frustrating and time consuming than it had to be.

1. None of you guys logged your tests. I don't know if it's because it was not directly highlighted during our meeting but it's very clear in the list of files that you were supposed to submit that a log of your console after testing be copied and properly named in a text file. I did the tests for you for now but whatever you wrote down in the testing excel sheet is what's present in your actual tests so if you failed to write the actual input, then that's what showed up. For next time, make sure to give me a different set of tests that won't completely kill the terminal as our tests will come one after the other to test a whole run-through of all the modules.
2. Especially moving toward the optional v2 and v3 submissions, absolutely no late submissions. If you fail to submit v2 and/or v3 by the agreed upon submission date and have failed to communicate to the entire group in the general chat why and when to expect it, I will take it as a sign that you don't want to do further versions.
3. Update your actual time spent in the project plan.
4. Reread the FOUR (4) files expected of you



Appendix B Deliverables and Deadlines

Version 1 "C+" maximum marks for highest quality (Submission marked out of 100% for quality × .694)

The term *module* below applies to any of the modules named in Appendix A.

1. Each group member's unit module:
   a. Note: "*module*" in filename does NOT include any indication of the version.
   b. *module*.h – code entered, commented.
   c. *module*.c – Version 1 code entered, commented, compiled.
   d. *module*_test_cases.xlsx – comprehensive unit tests to be run, with a record of post-test results.
   e. *module*_testing.txt – console text captured *as text* showing unit test inputs and outputs. Image capture, e.g. PNG, files will **not** be accepted.
   f. Upload above four files to your group's MS Teams channel in the **root of Files**.
   g. Update the Project Plan with group member's actual hours and all tasks' status at time of upload.

For the want of a nail the shoe was lost,

For the want of a shoe the horse was lost,

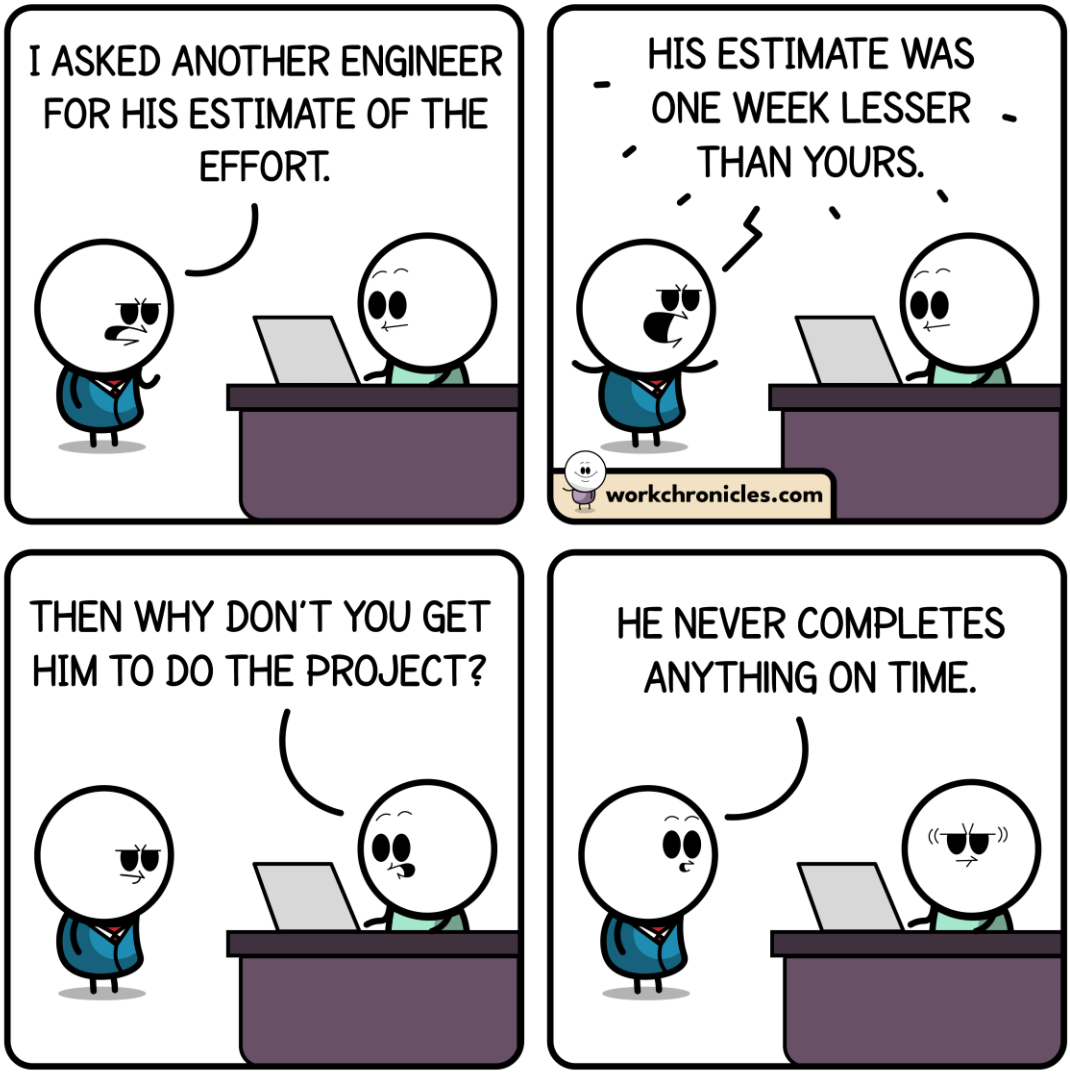
For the want of a horse the rider was lost,

For the want of a rider the battle was lost,

For the want of a battle the kingdom was lost,

And all for the want of a horseshoe-nail.”

— Benjamin Franklin



👋 **Like the comic? Don't forget to follow.**
Find Work Chronicles on IG, Twitter, LinkedIn, Reddit.

Work Chronicles
workchronicles.com