**Coursework Submission**

This is an evolving document, this version 06/12/18 09:54:16.

**Proposal**

We drop "submit" from 2019/20 academic session. We replace with multiple alternatives depending on per-course teaching/submission and administrative requirements. We strongly advocate one default submission mechanism.

**Reasoning**

The "submit" programme currently used by the majority of our courses was written over 20 years ago, but crucially has only been barely maintained since that point. While it does meet quite a lot of our needs it is not a robust code base. For example, in the last year two security issues have been identified that could have been trivially exploited to give any student full access to all submissions. The command is only available on the DICE command line – making it harder for external students on courses and DLAS students. The command is used for submission in online examinations and it is very hard to see any alternative (external system) being as usable or robust in that context. OTOH for the purpose of online examinations its functionality in this context could be entirely replaced by a very short Python script for example. Some argument can be made that use during semester will help familiarity during the exam – this is true but the usage is simple enough, students already have to use an unfamiliar command to get the exam paper itself, also they can attend the mocks so I don't think this argument really holds any weight.

**Comments**

We are currently looking from feedback from students and academics regarding this proposal.

**Requirements**

Below is an initial list of requirements for a submission system by Paul Anderson. They are specific to his needs and courses but nonetheless we think largely apply across the board. More suggestions from other academics, including course specific requirements, welcome.

Please identify which of Paul's suggestions you feel are critical, desirable, or not required (as they relate to your own courses). Please add your own suggestions to the bottom of the table. These will then be collated and shared more widely with all teaching staff in the School.

Below is a summary of responses, as received by 13 March, from course organisers responsible for courses with a cohort >200.

| Requirement | |
|---|---|
| Anyone with an EASE account should be able to submit any work. | |
| Critical | X |
| Desirable | X X X X |
| Not required | X |
| Submission should be possible from anywhere. | |
| Critical | X X |
| Desirable | X X X X |
| Not required | |
| Submissions should be accepted in a format determined by the individual assignments. | |
| Critical | X X X X |
| Desirable | X |
| Not required | X |
| It should be possible to put a per-assignment size limit on the submissions. | |
| Critical | X |
| Desirable | X X X X X |
| Not required | |
| Submissions should be possible at any time. | |
| Critical | X X X X X X |
| Desirable | |
| Not required | |
| There should be a persistent and stable URL for the submission of each item of work. | |
| Critical | X X |
| Desirable | X X X |
| Not required | |
| A receipt or check on the submitted files is sometimes useful. | |
| Critical | X X |
| Desirable | X X X |
| Not required | X |
| Staff should be able to download all of the submissions using a scriptable command line. | |
| Critical | X X |
| Desirable | X X X |
| Not required | |

| | |
|---|---|
| **Any configuration for the system (deadlines, required file, size limits, etc) should be possible by supplying a simple text file in a standard format. A web interface for editing this might be an optional extra.** | |
| **Critical** | **X** |
| **Desirable** | **X X X** |
| **Not required** | |
| **It is useful to remind students about "Good Scholarly Practice" when they submit their work.** | |
| **Critical** | |
| **Desirable** | **X X X** |
| **Not required** | **X X** |
| **Some stats are a nice extra.** | |
| **Critical** | |
| **Desirable** | **X X X** |
| **Not required** | **X Nice but far less important than others** |
| **Some people are using Git for submission.** | |
| **Critical** | |
| **Desirable** | **X X** |
| **Not required** | **X X X** |
| **Be able to deal with pair / group work** | |
| **Critical** | **X X** |
| **Desirable** | **X** |
| **Not required** | **X X** |
| **Easily deal with late submissions**<br>Be able to draw down only late submissions. | |
| **Critical** | **X X X X** |
| **Desirable** | **X** |
| **Not required** | |

**Additional requirements:**

- **Run post processing on submitted solutions.**
- **Absolutely critical:** The biggest problem with our current submission system is that students cannot check the files that they have just submitted to make sure they submitted what they thought they did. This is an absolutely critical feature for any new system. At the moment lecturers have to deal with individual "mistakes" and this is a big source of potential misconduct/unfairness.
- **If marking is part of the system,** then it is critical to be able to do a bulk upload via .csv file or similar. No web forms/hand entry!
- The ITO/SST should be able to record any extensions so that the submission system can compute late penalties. Late penalties should be provided in a format that can be

applied to the students' marks (in APT?) without error-prone and time-wasting re-entry.

- I really like the way submit exposes submissions on /afs so I can grab updates with rsync. Having everything in a standard VCS (probably git or svn) would be ok too.
- Critical features:
  (a) Submissions can be retrieved via a scriptable unix command line.
  (b) Late submissions must be clearly distinguished from normal ones.
  It must be possible, via a scriptable unix command line, to retrieve the latest ordinary submissions and separately to retrieve the late submissions.
  (c) The interface must not give the impression that late submissions are OK by default. Deadlines are set for a reason, and exceptions are exceptions that require special permission.
  (d) Submitter certifying following rules about the submission (scholarly practice, size and formal requirements), e.g., by tick-boxes.
  This is critical for dissertations submission, less so for progress reports.
  This is currently handled via the dissertation submission webform; see
  http://www.inf.ed.ac.uk/teaching/courses/diss/thesis_instructions.html
  https://projsubs.inf.ed.ac.uk/cgi-bin/submission_form_msc.pl
  Desirable features:
  - Submitter can update submission before the deadline.
  - Specify reasonable upper bounds on file sizes of submissions.
  Generally, I am worried about the potential size and complexity of any new fits-all submission system. The submit command has its limitations, but it is elegant in its simplicity and suffices for submitting the progress reports. If one has a large number of users, simplicity is a virtue.
  The webform for dissertations submissions at
  http://www.inf.ed.ac.uk/teaching/courses/diss/thesis_instructions.html
  https://projsubs.inf.ed.ac.uk/cgi-bin/submission_form_msc.pl
  implements the special requirements for this case (in particular
  many certification tick-boxes).
- First, seeing "Some students have never used the Unix command line, and are not required to." is kind of sad - any Informatics/Computer Science department should inject some Unix into its curriculum as an integral part, whether it is implicit (by submitting homework) or explicit (through a course).
- Second, it is important to note that students should test their code on DICE anyway, otherwise it becomes very difficult to test their code and adapt to each student's specific submission. We work under the assumption that students have to make their code work on DICE, otherwise it may be dependent on a specific installation of a piece of software on their computer, which we have no access to, and wouldn't want to install. So, just to reinforce - we always state to students - your code should run on DICE, perhaps even developed there, because that's the final environment in which your code is tested in. Otherwise, if it doesn't work, you may lose marks, even if it works on your computer.