

Coursework submission, marking and feedback

Proposal

We drop “submit” as the default mechanism for students submitting coursework 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.

Alternatives

The Learn assignment tool can meet many (but not all) requirements listed below. For standard submission of text based work, either Learn’s own assignment tool or Turnitin (accessed via Learn) will often suffice. Students on joint programmes will be familiar with these tools. However, we recognise that for code submissions, these tools are not a good match. We propose piloting other third party solutions in 2019/20. For example:

- [codegrade](#) (to be piloted by Volker Seeker on Inf1b) and
- [gradescope](#) (used by Milos Nikolic in 2018/19 and recently acquired by Turnitin. We are currently in negotiation regarding licensing).

We plan to evaluate these, alongside a more in-depth scoping exercise performed by one of Aurora Constantin’s 4th year UG students titled ‘Improving Learn through User Participation’.

A list of requirements for a submission system was initially drawn up by Paul Anderson. These were circulated to all teaching staff, who were invited to rate these as critical / desirable / not required as appropriate to their teaching context. Teaching colleagues were also invited to list additional requirements not identified by Paul. The results of this survey can be found below.

Submit Requirements: survey response

A suggested list of requirements for a coursework submission system was circulated to all teaching staff. Twenty Eight (28) responses were received. These are listed below. I have commented on each suggested requirement, listing how current, centrally provisioned services can meet (or not) these requirements.

Anyone with an EASE account should be able to submit any work

- **Critical** **10 (35.7%)**
- **Desirable** **13 (46.4%)**
- **Not required** **5 (17.9%)**

I think there's was some confusion over this. Paul Anderson had meant that any UoE student (even those not enrolled on the course) should be able to submit coursework. I'm not convinced this is how many interpreted the requirement. Those marking large classes (and indeed those not) are not expected to, nor I imagine would want to, mark coursework for students not enrolled on their course.

Anyhow, Learn allows only students enrolled on the course (ie, not those auditing the class) to submit via their assignment tool.

Submission should be possible from anywhere

- **Critical** **11 (39.3%)**
- **Desirable** **15 (53.6%)**
- **Not required** **2 (7.1%)**

The Learn assignment tool supports submission via web browser. Students can submit via most devices (desktop, laptop, mobile device such as tablet) as long as they have access to the internet. If working on a network with slow upload speeds, they should ensure they have left enough time prior to deadline to submit.

Submissions should be accepted in a format determined by the individual assignments

- **Critical** 14 (50%)
- **Desirable** 12 (42.9%)
- **Not required** 2 (7.1%)

Learn supports students uploading multiple files, of any file format. There is no way of ‘forcing’ students to submit in a particular format.

Turnitin can require students to submit in a ‘supported’ format – ie one which can be used to run through their originality checker.

It should be possible to put a per-assignment size limit on the submissions

- **Critical** 1 (3.6%)
- **Desirable** 16 (57.1%)
- **Not required** 11 (39.3%)

Learn does not force an upload limit. Files of several GB can, and have been submitted.

Turnitin has a file upload size limit of 60MB.

Submissions should be possible at any time

- **Critical** 14 (50%)
- **Desirable** 12 (42.9%)
- **Not required** 2 (7.1%)

The instructor chooses the Learn assignment submission window. The assignment, by default, becomes available once the assignment is created. A deadline is entered, but students can submit after deadline.

Turnitin can accept submissions after deadline – but only if the student has *not* already submitted prior to deadline. In cases such as these, an instructor on the course is required to delete the earlier submission.

There should be a persistent and stable URL for the submission of each item of work

- **Critical** 8 (28.6%)
- **Desirable** 16 (57.1%)

- **Not required 4 (14.3%)**

Learn does not support this.

A receipt or check on the submitted files

- **Critical 16 (57.1%)**
- **Desirable 11 (39.3%)**
- **Not required 1 (3.6%)**

A receipt is generated for the student on submission. However, no check is performed on the submitted files against preselected criteria.

Staff should be able to download all of the submissions using a scriptable command line

- **Critical 17 (60.7%)**
- **Desirable 9 (32.1%)**
- **Not required 2 (7.1%)**

Not available presently. Tim Colles to look into this. (Staff can download all of the submissions via the web browser.)

Any configuration for the system (deadlines, required file, size limits, etc) should be possible by supplying a simple text file in a standard format

- **Critical 7 (25%)**
- **Desirable 16 (57.1%)**
- **Not required 5 (17.9%)**

This is not supported by Learn. Deadlines are entered into a date and time field when creating the assignment in the browser.

It is useful to remind students about "Good Scholarly Practice" when they submit their work

- **Critical 2 (7.1%)**
- **Desirable 15 (53.6%)**
- **Not required 11 (39.3%)**

This is usually performed in other schools by using the 'adaptive release' feature. Students are required to 'sign' a declaration of own work before the submission box is revealed.

Reports / stats on submissions

- **Critical** 4 (14.3%)
- **Desirable** 17 (60.7%)
- **Not required** 7 (25%)

Reports can be run to show submission times (useful for the ITO when reporting on late submissions). Grade Centre data can be downloaded as a csv file, allowing reports to be generated on grades received. Data fields include name and student ID. Calculated columns can be created in the Grade Centre. If used, these should mirror the assessment structure in APT so students can easily see and estimate their final coursework grade.

Be able to deal with pair / group work

- **Critical** 10 (35.7%)
- **Desirable** 7 (25%)
- **Not required** 11 (39.3%)

Learn has a 'group submission' option within its assignment tool. Students can be grouped using the group tool. When the group submission option is enabled, only one member of the group needs to submit the coursework on behalf of the group. This is marked and the mark is automatically cascaded to all members of the group.

In addition, there is a peer assessment tool called WebPA. This allows students within a group to assess their own contribution and that of their peers. This then creates a multiplier which is applied to the group mark, thus giving each member of the group an individual mark.

Easily deal with late submissions (It must be possible, via a scriptable unix command line, to retrieve the latest ordinary submissions and separately to retrieve the late submissions)

- **Critical** 13 (46.4%)
- **Desirable** 13 (46.4%)
- **Not required** 2 (7.1%)

Not possible via command line in Learn. Instructors can download selected user submissions, all attempts, and all last attempts.

Run post processing on submitted solutions

- **Critical** 3 (10.7%)
- **Desirable** 14 (50%)
- **Not required** 11 (39.3%)

All submissions can be downloaded at the same time via the Grade Centre. Each downloaded file contains student ID (if anonymity is disabled) or Attempt ID (if anonymity is enabled), date and time of submission. And post processing would need to be run after download.

Students can check their submission has been successful

- **Critical** 22 (78.6%)
- **Desirable** 4 (14.3%)
- **Not required** 2 (7.1%)

Students receive an email confirming successful submission of their coursework. This is true for both Learn assignments, and for Turnitin assignments accessed via Learn. We should ensure a clear and consistent message is given to all students, across all courses, confirming this and encouraging them to check spam mail etc. Both Learn and Turnitin also allow an instructor on the course (usually the course secretary) to email all non-submitters. This is a useful tool but should be applied consistently, at the point of deadline.

Submitter must certify rules about the submission (scholarly practice, size and formal requirements), e.g., by tick-boxes

- **Critical** 5 (17.9%)
- **Desirable** 11 (39.3%)
- **Not required** 12 (42.9%)

This can be achieved via the 'adaptive release' function within Learn.

Submitter can update submission before the deadline

- **Critical** 16 (57.1%)

- **Desirable** 7 (25%)
- **Not required** 5 (17.9%)

Learn can accept a single submission, a limited number of submissions (instructor chooses the cap) or unlimited submissions. Each submission carries its own date and timestamp and the marker can choose to mark a particular submission or the last submission for each student.

Turnitin allows resubmissions up until the point of deadline. These overwrite previous submissions.

Free text entry responses to the survey

For projects, students are required to submit a directory of project-related material. There are sometimes problems with submitting this, because it can be a lot of data and lots of students are submitting at the same time. I'm working with Tim Colles on a solution that allows the possibility of submitting a hash of the directory at project submission time, and then submitting the directory itself later. The hash allows us to check that the directory hasn't been modified since the hash was submitted, before the deadline. Something along those lines seems like a good way of dealing with that issue.

Multiple students submitting large amounts of data at the same time should not be an issue using centrally provisioned services such as Learn.

Ideally, it should be possible to run tests on the submission at submission time, and reject the submission if the tests fail.

Something like this can be achieved using the [Notable service](#) (linked to via Learn). However, this is not possible using the Learn assignment tool.

Keep all submitted copies - (extensions may be agreed after submission)

Learn can accept unlimited submissions, which are kept for several years.

- * Does "post-processing" include running a validation script on the submitted file, and rejecting / reporting to the submitter if invalid? (for example to check that a source file does not contain any changes outside of specified regions that were to be filled in)**
- * Currently, submit accepts a single file, which means that assignments involving multiple files / larger source tree need to be zipped up by submitters. It might be convenient to allow multiple files to be submitted at once, through a single form. Not sure how common this scenario is.**

Multiple files can be submitted at once, via a single web form. However, and 'post-processing' would need to be handled outside of Learn.

System should be able to handle multiple files of different types (e.g. program files, reports, data files). E-mail to course organiser would be nice: E.g. at deadline and at any late submissions.

If pair/group work is submitted then it would be nice if the system should require all relevant UUNs and avoid double submissions. It could be an idea to check the systems like easychair for design features.

Learn can accept multiple files of different types. The instructor can receive an email when submissions have been made via their Learn notification settings. Group work can be managed as detailed above.

Group submission where the students define the groups

Submissions are visible to the TA as they come in and can be marked before the deadline.

Students can sign themselves up to their own groups. As above – any instructor on the course (including TAs) can receive an email when submissions have been made via the Learn notification settings. It is inadvisable to mark prior to deadline to prevent any redundant marking.

- * The design and implementation (or adoption) should be subject to passing a SOFTWARE SECURITY REVIEW -- this seems obviously critical, given this is an argument for moving away from current system, we want the new system to have better security.**

*** Another consideration for security and functionality is an AUDIT TRAIL and perhaps HISTORY of past submissions. In the past, students have complained that the submit system has lost submissions (possible race condition at one point) and also made mistakes by submitting broken versions over good ones, etc. At the least the system should keep a careful log of submissions, not just a receipt/snapshot of current submissions. Better would be a history.**

*** A particular design idea (linked to above) would be to allow students to use standard version control systems to submit their software-related work, e.g., use a git repository. This would offer many advantages including integration with workflow tools, browsing, etc.**

Learn keeps all submissions for the lifetime of the course instance (currently >7 years). [GitHub Classroom](#) is definitely an option and is used elsewhere in the University.

I always use TurnItIn for text submissions; I'm happy to consider an alternative if it includes detailed plagiarism check data.

There is no plans to move away from Turnitin if it works for your course. Turnitin remains a centrally provisioned service.

CT is a little bit special since my current way of dealing with student submissions is to pull "master" from their gitlab (University hosted) repository at a specific time (at the deadline). If we wish to allow students to submit whenever they want, what I guess could work well for my specific course is for student to submit a commit id from their gitlab repo.

This makes me think that a very general approach to submitting coursework could simply be to have all student use the University gitlab service for coursework and the submit command would simply consists of sending the commit id the student wish to submit. I would then imagine the command would then automatically clone the gitlab repo (with the specific commit id) and store it somewhere, and notifying the student that the submission was successful.

The biggest advantage of this solution is that the marker can access the whole history of what the student did (this could be very useful for investigating cases of plagiarism for instance).

As mentioned above, [GitHub Classroom](#) is definitely an option and is used elsewhere in the University.

Re "scriptable command line" it depends what the alternatives are.

Alternatives in Learn are download all submissions via a button in the administration area of the Learn course.

Please run the existing submit system alongside the new system for at least one academic year so that academics have the choice to use either the new system or the existing one. The existing submit system has several strengths which a web-based system might not have (including supporting submission of large files).

For the benefit of lecturers on first semester courses, the new submission system should be available at the start of the summer break to allow testing of the system before it is used in first semester. Having the new system made available at the start of the semester would be too late.

To be discussed. I believe Tim want to retire Submit summer 2019. If we continue to support submit beyond this time, there is an argument that there is no encouragement to move and we simply delay the decision for another 12 months. Various Informatics courses have been using centrally provisioned submission systems already.

(Context: the course I currently teach does not use submit for coursework, but does use examsubmit for submission of exam-condition lab exercises. In my answers I'm mostly thinking about examsubmit, therefore. But of course I have opinions based on courses I have taught or might teach, too.)

It would be desirable to be able to configure whether submissions are available to lecturers by student number, or by something more anonymous, e.g. exam number. Currently, programming exams are made available by student number, and we rely on markers' integrity to refrain from deanonymising, which they trivially could. While I find it no real temptation to deanonymise (and these days my memory for numbers is not so good that I accidentally remember any student ids!) it would be better to be able to tell students that, as with paper exams, the marker cannot know who they are while they mark. However, this should not be the only mode in which to submit work, because the more we treats students as numbers the more they feel like numbers - often, it's a

positive thing if the marker knows who the student is. Ideally it would be easily configurable.

Learn can support anonymous and identifiable submissions. Anonymous submissions are entirely anonymous, with no identifying information available to any instructor on the course (including the ITO).

Given that many course would benefit from using code revisions (including mine) I strongly advocate for using GitLab. We already have a license if I am not mistaken and it should be easy to make this work for different courses.

As mentioned above, [GitHub Classroom](#) is definitely an option and is used elsewhere in the University.