Summary and response to consultation

Sharon Goldwater

5 Sep 2018

1 Executive summary and recommendations

36 teaching staff commented on the proposal. Adding the 7 members of the curriculum group and ∼5 others consulted at earlier stages who didn't respond at this stage, a total of ∼48 staff had input. This section summarizes the main feedback from the consultation phase and provides recommendations. More details of the feedback are in Sec 2.

1.1 Main take-aways

1) Of the 8 guiding principles, our current curriculum is widely seen as not achieving four of them: programming skills; integration of social/ethical/professional issues; simple structure; deliverable within resourcing constraints. On aggregate, respondents were optimistic about the proposal being likely to achieve all eight of the guiding principles.[1] (For plots, see Section 2.3.)
2) There is broad support for increasing the time devoted to programming, and for increasing resourcing to support this. Many said the increased time for programming is one of the main strengths of the proposal, and no explicit objections were raised about the proposed changes to UG1 that help implement the increases (but see (4).
3) Respondents recognize that the continued move towards 20pt courses will reduce student flexibility, but most who commented view this as an acceptable trade-off and several said that streamlining is one of the best aspects of the proposal.
4) Amongst those whose comments were generally positive (15), brief/noncommittal (11), or somewhat skeptical (3), the most common concern was resourcing the changes and the more practically oriented courses. Reduced intellectual diversity in pre-honours was also somewhat concerning for many.
5) Of the 8 respondents who were strongly critical, all of them were deeply concerned about the reduction of theory and/or AI in pre-honours (especially ug2 courses).
6) Most respondents feel that project clusters are a good idea in moderation (i.e., not for all students/projects), and some would consider doing it themselves.
7) There is a strong base of staff who are potentially willing to teach the proposed courses related to algorithms, discrete maths, and data science, though many pointed out that staff in data science are already oversubscribed.
8) Despite the broad support for more programming, very few respondents were even willing to consider teaching the traditional programming/SE courses: Inf1b (2), Inf2 SE (2-3), ILP (3), SDP (2).

[1] This might reflect over-optimism for the untested vs. known weaknesses of the current curriculum.

1.2 Recommended actions

Unless otherwise noted, actions are on SG (or to be delegated by SG).

Relevant to ug1:

1) Plan to go forward with the change to Inf1b for 2019-20. (V Seeker and P Anderson have begun preparing a detailed course proposal, in consultation with S Goldwater.)
2) **(StratComm/DoT/HoS)** Make sure (at least one) permanent position becomes available for a Grade 8 University Teacher within the next year, to help ensure retention and career progression for existing UTs and to make UT positions more attractive to future hires. Consider UTs and other creative ways to help develop and staff programming/SE courses.
3) Include an explicit commitment in Inf1b (and Inf1a if not too late) to include illustrative examples that reflect unique Edinburgh topics.
4) Develop a series of presentations to run occasionally during ug1, with two goals: (1) introduce students to greater social and intellectual diversity within Informatics, (2) develop a stronger learning community amongst students, and between students and staff. Lecturers would be encouraged to discuss aspects of their research interests as well as their personal story or advice, and students would be encouraged to help plan the series and provide questions or topics. (The ug1 year organizer and student support officers are piloting a one-off version of this to occur during induction week this year.)

Relevant to ug2/3:

1) **(CRC/HoS/staff)** Consider alternative proposal for compulsory courses (see paper), which aims to alleviate the concerns of the most negative respondents.
2) Move forward with some of the proposed course mergers amongst non-compulsory courses, to begin to streamline the upper-level curriculum. (S Goldwater and others have begun speaking to the lecturers involved, aiming for course proposals this autumn.)

Relevant to ug4:

1) Arrange a session with MSc project cluster supervisors (e.g., Leather/Seeker, Goldwater, Murray (?); plus Barbara) in late Aug to see how these went and begin to formulate best practices.
2) Plan to try adding a few UG4 clusters for next year. Further discussion of logistics and marketing to follow meeting above.

Relevant to all years:

1) **(DoT)** Continue to develop new resourcing model for approval by StratComm.
2) **(DoT/StratComm)** Encourage ug1 lecturers to consider ways to increase teaching support for programming, permitting greater use of resources if need be, but also emphasizing efficiency. Consider more broadly how larger numbers of ug students can be trained and utilized as teaching support.
3) Propose a more detailed transition plan toward the new curriculum. (An initial version of this has been developed, undergoing discussion and refinement with DoT/HoS; can summarize verbally).

2 Further details of consultation feedback

2.1 Neutral/positive respondents

15/35 respondents gave comments that were clearly positive about the substance of the proposed changes, with 11/35 writing either few comments, or comments that did not show a clear overall positive or negative view.

Of these respondents, the most common positive aspects cited were:

- more time for programming and/or projects
- reduced redundancy/more efficient
- updated/more modern curriculum, better alignment with ACM guidelines.

The most common concerns were:

- Resourcing. Nearly everyone mentioned this in one way or another. The most common points were: (a) the effort to transition will be significant, do we have enough will and resource to do it given that we are already overstretched? A few people wondered if the changes are enough to be worth it. (b) The proposal seems more streamlined than our current curriculum, but more practically oriented courses likely require more support, and the compulsory courses don't make use of staff interests as well as at present. So it may actually be harder to deliver well than at present.
- Intellectual diversity in pre-honours. While this group felt that overall the proposal was an improvement, many expressed the desire to ensure that the courses maintain intellectual diversity and an Edinburgh flavour, such as by explicitly committing to including examples from our various research/application areas.
- The DRW course in particular, which several people felt was less well-specified and less coherent than others.
- Requirements in ug3. Several people felt that the requirement for an additional systems course was too much.
- Some concerns about reduction in theory (mainly formal language theory) but much less strong than in the negative group.
- One person who was generally positive said the proposal could be more radical, but didn't specify further. A couple of more negative responses echoed this, one saying the proposal seems like a lot of shakeup for not much substantial change.

2.2 Negative respondents

Eight respondents had severe criticisms for the proposal. Nearly all of these boil down to saying that the proposal has insufficient coverage of theory and symbolic AI and too much systems and data science. It's not clear if everyone agrees on which parts of theory—some mentioned algorithms, some mentioned logic, some mentioned theory of computation, and some were not specific.

Specific criticisms in this vein include:

- The pre-honors curriculum is weak in areas where we have key strengths and requires too much teaching needed in areas that we either don't have many staff or where staff are completely overloaded already.
- The pre-honors curriculum doesn't provide a good enough roadmap of possibilities in later years.

- The graph theory/algorithms that would be cut from DMMR is not actually covered in IADS. [However the proposal says that these could be included in DRW; I think a lot of people didn't read the details that mentioned this.]
- We should not be aiming toward the ACM standard; this is a template for mediocre universities to aspire to.
- More theory should be incorporated into ug2 rather than having a "theory option" in ug3; it's too restrictive to force all theory courses to be 10 credits.
- ug3 should not require any additional systems courses.

Additional points included:

- There isn't enough time in DAIT to cover all the topics and skills listed, especially including good enough coverage of probability/statistics.
- Despite the attempt to reduce redundancy in teaching of probability theory, some students are still being told to do it twice (in DAIT and PwA).
- DRW is underspecified and/or incoherent, or would be more appropriate in upper years.

2.3 Achievement of principles

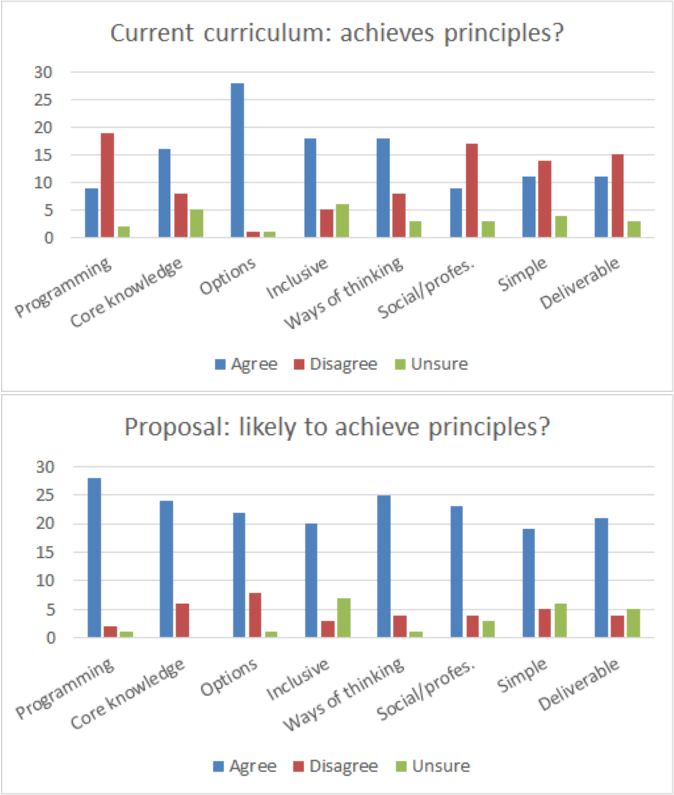

Figure 1: Number of respondents who (strongly or somewhat) agree/disagree that the current curriculum achieves each principle (top) or that the proposed curriculum is likely to (bottom).

Respondents were asked to what extent they agree that the current or proposed curriculum adheres to the principles set out. The full wording of the principles is repeated below as a reminder, with results of these questions in the plots above.

A1) All students (including those on joint degrees) should graduate with strong programming abilities, and confidence in their own programming abilities.

A2) All single-honours students should graduate with a solid grasp of core computer science and/or software engineering knowledge areas (as defined with reference to external recommendations, such as the ACM guidelines *Computer Science Curricula 2013*—henceforth **CS2013**.[2] Joint degree students (including Cognitive Science students) could miss out on a few of these but not too many.

A3) In addition to core courses, all students should have the opportunity to explore outside subjects as well as specialized areas within Informatics, to allow them to experience the breadth and unique flavour of Informatics at Edinburgh.

A4) The curriculum should aim to be inclusive of all students, regardless of previous programming experience, demographic factors, or special needs.

A5) Both individual courses and the curriculum as a whole should focus on *ways of thinking and working* as well as specific knowledge areas. That is, we should aim to develop critical “soft skills" (graduate attributes) such as: ability to communicate ideas, work in teams, manage time, and deal with uncertainty; general problem-solving skills, critical thinking and reasoning, and independent learning.

A6) Integrate discussion of social and professional issues throughout the curriculum, as relevant within the context of specific knowledge areas and applications.

A7) We should aim for a simple structure that reduces administrative overhead and includes constraints only where truly necessary.

A8) The curriculum should be deliverable within the constraints imposed by resource limitations (ultimately, the time available from lecturers and teaching support staff, and the time available to students).

2.4 Resourcing and projects

- Plenty of staff are potentially willing to teach data science-oriented courses: all aspects of these courses had 6-13 “yes” or “maybe” responses to the “would you consider teaching it?” question. However, these staff are also heavily in demand at MSc level.
- Mainly theoretical courses also had a strong base: DMMR (7), Inf2 IADS (10).
- Despite the broad support for more programming, very few respondents were even willing to consider teaching the traditional programming/SE courses: Inf1b (2), Inf2 SE (2-3), ILP (3), SDP (2).
- Remaining courses are in between: Inf1a (6), Inf2 CS (4). Only 2 said “maybe” for Computer Security, but no Security staff responded to the survey.
- 6 people said “yes” and 8 said “maybe” they would consider planning/supervising a project cluster. Comments suggest most people think clusters are a good idea in moderation, with a few very enthusiastic or very skeptical. Common concerns raised were: this won’t work for all areas; we need to make sure it’s attractive to students; we should still make sure students have the option of individual projects. Other concerns were: how would we ensure academic honesty without individual meetings?; students won’t learn as much as from 1-1 meetings where much of the learning in the degree happens.

[2] AI students are an exception: there is considerable overlap in core knowledge between AI and CS/SE, but CS2013 is not directly applicable to AI degrees. We know of no external guidelines specifically for AI.