Course Proposal Form

Please see Page 2 for instructions on which parts of this form to complete, whom to consult with to avoid unnecessary effort, and where to send the completed form.

Proposer(s): **Date:**

Cover page: Basic permanent course information

Unless otherwise noted, items in this section are entered into EUCLID and **cannot** be changed without creating an entirely new course.

Course Name	Software Engineering and Professional Practice
Course Acronym *(used by the School only, e.g., for the Sortable Course List)*	SEPP
Course Level If the course is **only** available to MSc students, then it must be classed as Postgraduate. All other courses, regardless of level, are Undergraduate.	_X_Undergraduate __Postgraduate
Normal Year Taken	__UG1 _X_UG2 __UG3 __UG4 __UG5 __MSc
Also available in years *[This can be changed later if need be.]*	__UG1 __UG2 __UG3 __UG4 __UG5 __MSc
SCQF Credit Level Level 8 should normally be used for pre-honours courses. Level 10 should normally be used for optional UG3 courses (so UG4 students may also take them) and for courses aimed mainly at UG4 students. Level 11 should be used for courses aimed mainly at MSc students, whether or not UG4 students can also take them.	__7 _X_8 __9 __10 __11
SCQF Credit Points	__10 _X_20 __40 __60 __80 __Other:
Delivery Location	_X_Campus __On-line Distance Learning
Course Type	_X_Standard (default) __Dissertation __Online Distance Learning __Other (specify: Placement, Student Led Individually Created Course, Year Abroad)
Marking Scheme By default, courses use a numerical marking scheme. If you wish to use a grade-only marking scheme, your course proposal below should justify this.	_X_Standard (numerical) __Letter grade only

Guidance for remaining sections:

For an initial course proposal, please complete the **cover page and Section 1 (Case for Support)**, which asks you to describe the need for this course and to provide an overview of the course design, including the learning outcomes. **Please discuss your plans as early as possible with the head of Curriculum Review to avoid unnecessary effort.**

Send the form with these sections completed to the BoS Academic Secretary and head of Curriculum Review (listed on the BoS page) to obtain their comments before filling out the remainder of the form.

If a full proposal is invited, please complete the remaining sections and send to **iss-bos@inf.ed.ac.uk**.

2. Student-facing course description and additional feedback and assessment information.
This section provides most of the information students see in the DRPS entry for this course, as well as related details for BoS consideration.

3. Further information for BoS consideration: sample materials.

4. Additional Course Details required for DRPS. *[Administrative information such as delivery timing and prerequisites.]*

5. Placement in degree programme tables. *[Required for all level 9-11 courses; used to determine where the course will be added to existing degree programme tables.]*

6. Comments from colleagues. *[All course proposal should be sent to relevant colleagues in the area as well as to the appropriate year organizer and BoS Academic Secretary for comment in good time before the BoS meeting. Use this section to indicate what feedback has been solicited and received.]*

Colour coding and item-by-item guidance:

Guidance is provided in italics for each item. Please also refer to the guidance for new course proposals at http://www.inf.ed.ac.uk/student-services/committees/board-of-studies/course-proposal-guidelines. Examples of previous course proposal submissions are available on the past meetings page http://web.inf.ed.ac.uk/infweb/admin/committees/bos/meetings-directory but note that the proposal form was updated in Jan 2019.

Sections in gold are for student view and are required before a course can be entered into DRPS. You must complete these sections even if your course has already been approved based on other documentation.
Sections in orange are for School use but are still required for all courses (even those that have already been approved based on other documentation).
Section in gray are for consideration by the Board of Studies. They are normally required for all new course proposals but may be omitted in some circumstances (e.g., for invited course proposals) if you obtain permission in advance.

1. Case for support

This section is for consideration by the Board of Studies. The final two boxes (Learning Outcomes, Graduate Attributes) will also go into the student-facing course description.

Overall contribution to teaching portfolio and relation to existing curriculum

Please explain (a) what motivates the course proposal (e.g. a previous course having become outdated/inappropriate, an emergent or maturing research area or new research activity in the School, offerings of our competitors) and (b) how it relates to existing courses and degree programmes (including any prerequisite courses). Every new course should make an important contribution to the delivery of our Degree Programmes.

Software Engineering and Professional Practice (SEPP) is a new 20-credit compulsory second year undergraduate course proposed for 2020-2021 semester 2 for Software Engineering, Data Science and some Computer Science degrees. It is intended to replace the current 10-credit Informatics 2C- Introduction to Software Engineering (Inf2C-SE) second year semester 1 course, one of the most important motivations being including an earlier focus (both theoretically and through practical engagement) on professional issues in the curriculum, as required by most recent curriculum updates. Building on the experience and feedback results for the current Inf2C-SE course, but also on the requirements of the Software Engineering job marker, other important motivations for SEPP include the need for:

1. More up-to-date considerations of contemporary iterative development and deployment lifecycles
2. Confronting students more with the context of developing large software systems (apart from professional issues mentioned above, also e.g. issues of costs, tight deadlines, respecting non-functional requirements, unpredictability) and its implications on software development decisions
3. More emphasis on developing team working skills, but also the management of team work through e.g. version control
4. More practical experience with all of the above through engagement with a larger, realistic, case study
5. A consideration of pedagogical approaches which can improve understanding and foster deep learning: formative assessment by instructors and through peer review, self-assessment, reflection on feedback, the writing of reflective blog posts.

The contents of the SEPP course need to follow on from the first year Inf1B-Object Oriented Programming course, and moreover they must lay the foundations for the later Software Design and Modelling (SDM) and Software Testing (ST) optional courses. For these reasons, in preparing this proposal, we consulted with former and current lecturers in our school's Software Engineering courses, as well as with the current Director of Teaching and Deputy Directors of Teaching, administrative, learning technology and library staff members. This was achieved through both informal and, lately, an Edinburgh Learning Design Roadmap (ELDeR) 2-day workshop. The plan outlined in this proposal was the result of this workshop. In particular, through discussion with the current Inf1B course organiser, it ensures that SEPP will offer a natural progression from Inf1B by allowing students to practice their programming and test their understanding of good object oriented design. Additionally, students will be engaged with the management of a whole contemporary software development lifecycle and the challenges that it brings for the building of a larger scale software system as part of a more complex project.

Target audience and expected demand

Describe the type of student the course would appeal to in terms of background, level of ability, and interests, and the expected class size for the course based on anticipated demand. A good justification would include some evidence, e.g. by referring to projects in an area, class sizes in similar courses, employer demand for the skills taught in the course, etc

This course is aimed at second year students of Informatics programs who have passed the Inf1A- Introduction to Computation and the Inf1B-Object Oriented Programming courses. At the end of Inf1B, students will have familiarised themselves with object oriented concepts and Java programming. They will have been informally shown the motivation for an engineering approach to software development, and they will have a basic understanding of good engineering practice including version control, testing and documentation. SEPP will build on these initial foundations to shift focus on the engineering of larger software systems and the professional and other contextual issues involved. Implementation activities will also be carried out in Java, so no further programming background will be required from students. To help students revise their Java programming, the lecturers will provide links to resources and online tutorials (e.g. from LinkedIn Learning) on the Learn course page ahead of the first week of the semester.

Inf2C-SE currently has 236 students enrolled, however the demand for first year courses has considerably increased- to 400 students- this year. We therefore expect an increase in demand to over 300 for SEPP in 2020-2021.

Anticipated Resource Requirements

Estimate how much lecturing, tutoring, exam preparation and marking effort will be needed in steady state, and any additional resources needed to set the course up initially. Provide estimates relative to class size where applicable and discuss how support staff will be recruited and supervised, if the class is likely to be very large. Please mention any scaling limits due to equipment or space. If equipment is required, say how it will be procured and maintained.]

We do not expect the resource requirements for the SEPP (20-credit) course to be more than double those for the current Inf2C-SE 10-credit course. We propose the following:

- 3 lectures a week instead of 2 which is currently the number for Inf2C-SE
- No tutorials, so there will be no need for tutors (as opposed to Inf2C-SE where 21 tutor posts currently)
- For each student, one drop-in 2-hour lab with support from 3 demonstrators a week, starting in week 1. For 300 students, we could go for 5 lab groups (60 students each) a week and 13 demonstrator posts at 11 hours/post (to cover all semester weeks). The demonstrators will be recruited from teaching support staff with experience in Software Engineering and Java, prioritising former students on the Inf2C-SE course with very good results and those with a good amount of commercial experience as Software Engineers. They will be trained by the course organiser. The course lecturers will also act as demonstrators for 2 of the labs, and potentially switch these labs every once in a while to be able to provide good support to the demonstrators.
- For each student, one self-study 2-hour drop-in lab a week (provided to encourage meeting to work on assignment, no demonstrator support provided). For 300 students, we could go for 5 lab groups, maybe booked after hours every day of the week.
- We propose for this course to be coursework only (no exam), so no exam preparation time is needed
- One or two TAs at minimum 120 hours of work (60 hours each) will be required for preparing the assignment instructions, marking criteria, tests or auto marker.
- The marking is estimated to require around 180 hours (1 hour for each of 3 pieces of coursework for the 60 5-person groups), split into 6 marker posts at 30 hours each.

Quotas, special arrangements or unusual characteristics

Please specify if this course requires any special arrangements such as quotas or other registration arrangements; is a collaboration with another school or institution, or has other atypical characteristics that may affect finances or student registration. Further justification/information may be requested for such courses.

No.

Narrative description of the course aims and structure

Please describe the main goals of the course and how the course design will allow students to achieve those goals. This section should be consistent with the student-facing information provided below, but should provide additional information to help colleagues at BoS understand the vision and structure of the course. This description may refer to the learning outcomes and graduate attributes (next two boxes) and should explain how activities such as tutorials, labs, or in-lecture activities will support them, and how the proposed assessments will assess them.

For courses that are important pre-requisites for other courses, this section may also provide content/syllabus information which is too detailed for the student-facing description, such as a lecture-by-lecture syllabus.

Summary statement (from ELDeR workshop): This course will introduce the foundations of contemporary iterative software development and deployment lifecycles, emphasising hands-on experience, real-world large-scale systems, and professional practice.

To achieve these aims, we plan for assessment to be 100% coursework, to allow for more hours to be spent by students on hands-on experience with a larger-scale system. We do not believe that understanding of the practice and professional and ethical challenges of Software Engineering can be meaningfully assessed through examinations.

We plan to introduce a realistic case study that students would be working on in groups of 5. We plan to seek industry support for the topic of this case study. Allocation to the groups would be performed randomly, and additionally considering gender balancing, ahead of the first week by the ITO. Then, the week 1 demonstrator-supported lab would involve a team building exercise on producing a solution for a simple problem.

The case study, released in week 2, would include some skeleton code and clear interfaces to 5 separate modules, with each student in the 5-person group being required to work on one of these modules such that they could each be assessed independently (see submissions 2 and 3 below), and modules from other groups could be easily 'bolted in' for system testing purposes if any members leave the group. The team would be using git to commit all of their code changes. At the end of each week, they would need to integrate their modules, run tests and evaluate their solutions with other students playing the role of customers from other groups. From their results, but also to document their considerations of the different stages of development (including e.g. UML diagrams) and their management of the team, they would be required to write and submit as a group a reflective blog post at the end of each week, taking turns who writes it. The third week 1 lecture would include examples of good and bad such blog posts.

While in the first weeks their iterations would be based on knowledge from Inf1B- and likely involve delving directly into implementation- we expect students to start incorporating more and more of the Software Engineering concepts studied in lectures and through reading as the weeks progress. For example, we expect them to start documenting requirements and draw UML diagrams for design in subsequent iterations.

In week 4, individual students would be invited to peer review the code of colleagues from different groups implementing different modules.

The first coursework submission would take place in week 5 and have the purpose of formative assessment. Markers would offer formative feedback and provisional marks on the 5 blog posts to date, as well as run automatic tests on the system. General feedback on the coursework would also be provided in the week 6 lecture.

After week 5, some changes in requirements would be introduced to the case study to increase the difficulty of the modules and raise awareness about uncertainty in software development. The second coursework submission would this time be for summative assessment, and take place in week 7. The markers would consider how students have addressed their feedback since the first deadline, assess the new blog posts, re-run tests and check for student engagement from git commits and week 4 peer review reports.

After week 7, extensions to the modules may be required, and a consideration of professional issues would be required of each member of the group for coursework 3. For the latter, one idea is to introduce in the skeleton for the code opportunities for students to do something easily, but by doing so negatively influence their team mates or not exactly addressing customer requirements (and they could hide information from the customers to avoid making this clear). This would be a trap that students would need to avoid to act in an ethical way. In week 9, students would be invited for a new round of peer review. In week 10, they would be given access to the solutions on the same module by the other groups. The third coursework submission would take place in week 11, again for summative assessment. This time, apart from the blog posts and code, each student would need to prepare detailed documentation on their module, a discussion of professional issues and a comparison of other groups' module solutions with their own. Moreover, each group would be asked to submit results of system-level tests and a 2-minute video describing their solution.

To facilitate marking as well as self-assessment, a rubric-based marking scheme would be used by the markers. Moreover, we plan to provide such a marking scheme for the students to self-assess for the first two- formative and summative- coursework deadline. The markers can mark independently and then write a comparison of their provisional marks with the students', which can help students adjust their expectations regarding assessment and become better prepared for the future- summative- coursework. The students will be asked to reflect on their feedback with their new blog submissions.

Lectures are planned to include much more of a discussion of professional issues. We will also attempt to invite speakers from industry for guest lectures.

Summary of Intended Learning Outcomes (MAXIMUM OF 5)

List the learning outcomes of the course. These must be assessable (i.e., observable), so must specify what the student should be able to do concretely, not simply what they should "understand". Use concrete verbs that indicate (a) what type of assessment would be appropriate, and (b) what level of knowledge/thinking is expected (from recall to analysis to novel creation). ***Example verbs:*** *define, explain, implement, compare, justify. Assessments (described later) should be tied to the learning outcomes.*

Outcomes should typically focus more on the types of thinking/skills developed than on the detailed course content, and the level of thinking should be appropriate to the level of the course: outcomes for a Level 11 course should include more higher-level thinking skills than for a Level 8 course. Further guidance on writing learning outcomes can be found at https://www.ncl.ac.uk/ltds/assets/documents/res-writinglearningoutcomes.pdf

On completion of this course, the student will be able to

1) Explain the modern techniques used in the design and development of large-scale software systems

2) Apply and evaluate these techniques in a small-scale, but real life, scenario

3) Work effectively as part of a team

4) Analyse the professional and ethical implications of software engineering decisions and propose solutions

Graduate Attributes, Personal & Professional Skills

List the personal attributes and generic transferrable skills this course will help develop. Examples include

Cognitive skills: *problem-solving, critical/analytical thinking, handling ambiguity*

Responsibility, autonomy, effectiveness: *independent learning, self-awareness and reflection, creativity, decision-making, leadership, organization and time management, flexibility and change management, ethical/social/professional awareness and responsibility, entrepreneurship*

Communication: *interpersonal/teamwork skills, verbal and/or written communication, cross-cultural or cross-disciplinary communication*

On completion of this course, a student can be expected to have developed the ability to:

1. Work effectively as part of a team, including communicating and managing work in a group of developers, pair programming and code review
2. Interpret a vague system description, judge and weigh different system designs
3. Adapt to a small level of uncertainty in software development activities (through changing requirements)

4. Reflect critically on the team’s and their own progress
5. Reflect critically on professional and other contextual issues surrounding software development
6. Work to tight deadlines
7. Self-assess
8. Take feedback on board (from classmates acting as customers, peer reviewers, instructors), reflect and act on it
9. Interpret others’ code on both well known (i.e. same module) and less known problems (i.e. different modules)
10. Give constructive feedback on code written by others (as part of team work but also peer review exercises)
11. Present and communicate individual and team’s work

1. Student-facing course description and additional feedback and assessment information

Except where noted, all fields are required and will go into the DRPS entry for the course (for use by students). Important: any text in DRPS is effectively a contract with students, so should not include details that are likely to change from year to year.

Summary Description *Provide a brief official description of the course, around 100 words. This should be worded in a student-friendly way, it is the part of the descriptor a student is most likely to read. If this course replaces another course, please say so in this summary.*	
Keywords *Give a list of searchable keywords.*	
Course Description *A more detailed student-facing description of the course, which should normally include (a) a more in-depth academic description of the learning aims, nature and context of the course, (b) a rough outline of the content or syllabus, often as bullet points, and (c) a description of how the course will be taught, how students are expected to engage with their learning and how they will be expected to evidence and demonstrate their achievement of the intended learning outcomes.]*	
Assessment Weightings: *These should correspond approximately to the proportion of learning outcomes that each component assesses. More than 30% coursework requires specific justification.* *The expectation for a 10pt course is 20% coursework with the equivalent of one 15-20hr assessed assignment (but possibly split into smaller pieces). See 'components of assessment' below.*	Written Exam _____% Practical Exam _____% (*for courses with programming exams)* Coursework _____%
Further Assessment Information *Provide any further information that should go on DRPS for students. E.g., if the assessment includes required group work or if students must pass some individual component of assessment as well as the course overall.*	
Components of assessment and time spent on assignments (for BoS only) *If not already included in the course narrative description, please describe the type of assessments (oral presentation, report, programming, etc) and* ***how each component of assessment will assess the intended learning outcomes****. Where coursework involves group work, it is important to remember that every student has to be assessed individually for their contribution to any jointly produced piece of work.*	

Also estimate ***how many hours*** *students will spend on assignments. Please see the School policy on Workload and Assessment, which states that students should not be expected to spend more than 6-7 hrs/wk per 10 credits, including contact hours.* *Note that it often desirable to include formative assignments which are not formally assessed but submitted for feedback, often in combination with peer assessment.*	
Feedback Information *Provide a high-level description of how and what type of feedback will be provided to students, for inclusion in DRPS.*	
Additional Feedback Information (for BoS use only) *If not already included in the course narrative, provide further details on planned feedback arrangements. This includes how course feedback is solicited from the class and responded to, as well as what feedback students will get (either on work that contributes to their final mark, or not).* *The University is committed to a baseline of principles regarding feedback that we have to implement at every level, and the School encourages submission of at least one piece of written work for formative feedback.* *In general, formative feedback:* • *Should say how students can improve.* • *Need not be on individual work (e.g., consider a lecture or document summarizing common issues.)* • *Can include oral feedback during labs/tutorials* • *Can include feedback from peers* • *Clickers/TopHat/equivalents can provide in-class feedback for both students and lecturer.* • *Is returned in time for other forms of assessment to which it relates, to allow feedforward.*	

Breakdown of Learning and Teaching Activities

State how many hours students spend on each part of the course. The total should be 10 x course credits, but please also see the School policy on Workload and Assessment.which states that students should not be expected to spend more than 6-7 hrs/wk per 10 credits, including contact hours.

Assume 10 weeks of lectures slots and 10 weeks of tutorials, but these need not all be used. As a guideline, a 10-pt course typically has 18-20 lecture hours, but should have only around 15 lectures of examinable material; the rest should be used for guest lectures, revision sessions, introductions to assignments, etc.

Contact hours

Hours	Type
	Lecture Hours
	Seminar/Tutorial Hours
	Dissertation Project Supervision Hours
	Supervised practical/Workshop/Studio hours
	Feedback/Feedforward hours
	Summative assessment hours
	Revision Session Hours

Non-contact hours

Hours	Type
	Directed Learning & Independent Learning hours

Total hours:

Reading List/Learning Resources

You are encouraged to create resource lists using LEGANTO

1. Further information for BoS consideration: sample materials

A full proposal for a new course must include examples of exercises and assessment. Please provide these below, along with publicity information if the course is to be advertised outwith the School.

Course information and publicity *The course web page (typically the Learn landing page) will be linked from the Sortable Course List, and information such as timetables and assignment deadlines must be made available prior to the start of the academic year. Please specify here if any additional info/publicity is needed for your course: typically only if it is aimed largely at non-SoI students.*	
Sample tutorial/lab sheet questions *Provide a list of tutorial questions and answers and/or samples of lab sheets. These need not be fully fleshed out but should indicate what sort of exercises will be provided to help students learn the material.*	
Sample assessment materials *If the course is primarily assessed by **exam**, provide a sample exam question with model answers. Any non-standard exam format must be justified. The online list of past exam papers gives an idea of typical and alternative exam formats: http://www.inf.ed.ac.uk/teaching/exam_papers/.* *If the course is largely or primarily assessed by **coursework**, provide a sketch of a possible assignment with an estimate of effort against each sub-task and a description of marking criteria.*	
Any other relevant materials *Include anything else that is relevant, possibly in the form of links. If you do not want to specify a set of concrete readings for the official course descriptor, please list examples here.*	

2. Additional Course Details for DRPS

Except where otherwise noted, these fields are required for entry into EUCLID and will be visible to students in the DRPS entry.

Planned Academic Year of Delivery *(The first year you anticipate the course running, e.g. AY 2019-20)*	2020-2021
Course Organiser *(By default, the course proposer)*	Cristina Adriana Alexandru
Intended Delivery Period	___Semester 1 _X__Semester 2 ___Full Year ___Summer ___Other (please specify):
Timetable considerations/conflicts *For School use. Please specify any constraints to be considered (e.g. overlap of popular combinations, other specialism courses, external courses etc). Include whether the semester delivery is constrained or could be flexible.*	
Is this course available to visiting students?	__X_Yes (default) ___No **If no, please provide a justification here:**
Required pre-requisite courses *Use sparingly: these are enforced in PATH and can only be waived by approval from the School's Curriculum Approval Officer. Note that cross-year required pre-requisites may prevent MSc students from registering; consider using recommended pre-requisites or "other requirements" instead.*	___No _X__Yes (please specify full course name(s) and code(s)): Inf1A- Inf1B- Object Oriented Programming
Recommended pre-requisite courses	_X__No ___Yes (please specify full course name(s) and code(s)):
Required co-requisite courses *Specify any courses that must be taken in parallel with the existing course. Note that this leads to a timetabling constraint that should be mentioned elsewhere in the proposal.*	_X__No ___Yes (please specify full course name(s) and code(s)):

Prohibited Combinations *Specify any courses that may not be taken in combination with the proposed course].*	_X__No ___Yes (please specify full course name(s) and code(s)):
Other Requirements/Additional Information *This information is often used by MSc students and students from other Schools to see if they have appropriate background without having done our School's courses. So please avoid course titles, instead list specific knowledge and skills (such as mathematical concepts, programming ability or specific languages, etc).* *Also list any other constraints on registration, for example: "Only available to 4th Year Informatics students including those on joint degrees." or "This course is open to all Informatics students including those on joint degrees, and to students in the School of Mathematics. Other external students whose DPT does not list this course should seek permission from the course organiser."*	X___No ___Yes (please specify):
Visiting Student Pre-requisites	_X__Same as "other requirements" ___Different than "other requirements" (please specify):

3. Placement in degree programme tables: for level 9-11 courses only

This section is for consideration by the Board of Studies and will be used later by ITO to determine where the course will be added to existing degree programme tables.

Is this course restricted to students on a specific degree? *E.g., some courses are only available to students on a specific CDT or MSc.*	_X_No ___Yes (please specify and provide justification):
Is this course compulsory for students on any degree(s)?	_X_No ___Yes (please specify and provide justification):
Any issues for part-time students? *Normally, part-time students have access to the same courses as full-time students on the equivalent degree. If you anticipate any problems with this, please specify here.*	

For optional courses:

If this course is available but non-compulsory for students on various degrees (most courses), please fill in this section. The choices here determine where the course appears in degree programme tables (DPTs) and the 2-3 character tags are displayed in the Informatics sortable course list.

Should this course be tagged as 'ML' (machine learning foundations and methods)? *Courses with the ML tag are typically very high-demand and most degrees limit the number of ML credits. If your course might appeal to a similar audience but draw off students from these large courses, please select 'no' and choose one of the tags below.*	__No __Yes
If you chose 'no', please choose at least one of the following tags… *Ideally, select exactly one, unless there is a good argument for more than one. These three are used in various combinations for many of our degrees.*	__**FSS** (CS foundations, systems, and software) __**AIA** (artificial intelligence applications and paradigms) __**COG** (cognitive science: including HCI and NLP courses, but not most other AI courses. Please restrict to courses most relevant to natural cognition.)
…and also tick if any of the following tags or categories apply. *Do not tick any of these if you selected 'ML' already.*	__**NS** (natural systems: e.g., computation by or about biological or social systems. Many COG courses are also NS. This tag is mainly relevant for MSc in Informatics.) __**SE** (software engineering: including courses that are highly relevant to SE degrees. All SE courses should also be FSS. This tag is mainly relevant for UG SE degrees.) __Databases and data management systems (used for Data Science MSc and MSc(R))

	___Unstructured data and applications (used for Data Science MSc and MSc(R)) ___Level 11 Security courses (used for Security MSc)
If you are not sure which tags are most appropriate or have other questions about this section, please note any comments/issues here.	

4. Comments from colleagues

All course proposal should be sent to relevant colleagues in the area as well as to the appropriate year organizer and BoS Academic Secretary for comment in good time before the BoS meeting. Please indicate here what feedback has been solicited and received.

Additional Comments *Summarise any comments received from relevant individuals prior to proposing the course. If you have not discussed this proposal with others please note this.*	
Year Organiser Comments *Year Organisers are responsible for maintaining the official Year Guides for every year of study, which, among other things, provide guidance on available course choices and specialist areas. The Year Organisers of all years for which the course will be offered should be consulted on the appropriateness and relevance on the course. Issues to consider here include balance of course offerings across semesters, subject areas, and credit levels, timetabling implications, fit into the administrative structures used in delivering that year.]*	
BoS Academic Secretary Comments *Proposals must be checked by the Secretary of the Board of Studies prior to discussion at the actual Board meeting. This is a placeholder for their comments, mainly on the formal quality of the content provided above.*	