Professional Learning and Networking in Computing (PLAN C)

A submission to the Informatics Europe 2015 Best Practices in Education Award

Professor Quintin Cutts
School of Computing Science
University of Glasgow
Glasgow, UK
Quintin.Cutts@glasgow.ac.uk

Mr Peter Donaldson
Crieff High School
Crieff
Perth, UK
Peter.Donaldson@computingatschool.org.uk

This submission is made on behalf of the 50 lead teachers who have made the PLAN C project possible. If the submission is successful, the prize will be used to further support our endeavours and those of the lead teachers in their work with local teachers.

1. Description of the achievements

Summary

Professional Learning and Networking in Computing (PLAN C) is a Scottish Government funded project that has developed and delivered high quality professional learning to Scottish computing science schoolteachers. It follows the recommendations of Donaldson's influential *Teaching Scotland's Futures*[1] report on teacher professional development. The project consists of the following aspects:

- Based on the collation of a body of evidence on what works with respect to computing science teaching and why it works, we have operationalised the evidence by developing a series of novel approaches to teaching the subject.
- We have trained a network of 50 'lead' teachers drawn from secondary schools all over Scotland. The aim has been both to deliver the novel teaching methods, but also to develop a group of highly reflective teachers, able to identify and collate pedagogical content knowledge – the key to great teaching.
- We have supported the lead teachers to create 25 local teacher hubs spanning the country. Hubs meet approximately monthly and are led by the lead teachers following materials based on their own lead teacher training. At least 350 of the 650 secondary CS teachers in Scotland have been involved in the programme.
- We have run a series of additional events to enhance computing teaching at both primary and secondary level, some in conjunction with the national educational centre, Education Scotland. We have planned further activities, still to complete.

The independent evaluation of the project is showing that high-quality professional discussions are taking place in the local hubs, significantly deepening the teachers' understanding of core computing science concepts and enabling promotion and sharing of effective practice. Feedback from the classroom suggests a big improvement in pupils' uptake of the skills and concepts.

Background

Scotland has had national school computing qualifications for around 30 years, but these had become stale, and new qualifications, demanding significantly greater

[1] http://www.gov.scot/Publications/2011/01/13092132/0

programming and computational thinking skills, were introduced in 2013[2]. The existing body of around 650 teachers consists of a blend of deep enthusiasts through to those with a relatively poor grasp of computing concepts, reflecting the changing requirements for admission to CS teacher training; at one point, for example, computing degrees with little or no programming were acceptable. In the same year, 2013, the proposal for PLAN C was developed by a consortium of interested parties including particularly Computing at School Scotland (the grassroots association for computing teachers) and the Scottish Universities, with the aim of supporting all teachers to deliver these new enhanced qualifications. Funding was granted and the project has been led since August 2013, by the authors of this submission, a CS academic with a CS education research background and a highly experienced CS school teacher well-versed in the literature. The current funding for the project is due to end in August 2015, but there is clearly more work to be done and further funding is currently being sought.

The Scottish CS qualifications at each level (broadly for 16, 17, and 18 year olds) consist of two parts. These are SDD (software design and development) which mainly concerns programming, and ISDD (information systems design and development) which focuses on the web and databases as well as more theoretical areas such as systems, networking, security, privacy and so on. PLAN C has focused primarily on the development of computational thinking skills required for creating and understanding solutions in programming, web, and database languages, since it is these skills that are typically so hard to foster.

Novel, evidence-based, approaches to teaching and learning CS

Our approach to improving teaching and learning in CS can be summarised by the consideration of two fundamental and widely prevalent, almost universal, errors in the early stages of traditional computing teaching:

- **Computational systems, such as programming languages, are introduced by example**. Time is not spent ensuring that novices have learned how to talk about computational concepts, nor is it spent on teaching an explicit mental model of the various constructs they are using. Fundamentals, such as the difference between a formally defined language (e.g. a programming language) and a natural language, or elements of mechanistic reasoning such as the deterministic nature of machines, are not mentioned. These issues have been known about since Soloway's and du Boulay's writings in the 1980s and 90s, and reiterated by for example by Pea and Sorva, but little change has resulted from them in mainstream education. Instead, novices are left to develop their own mental models of a programming language's operation. From the so-called "alternative conceptions" literature, it is obvious that the number of possible mental models that can be derived from a small number of examples can be very large. For example, a typical assignment statement has several *viable* interpretations, any one of which might be intuited by a learner. Having introduced concepts by example, at best a teacher then spends most of their remaining time identifying and fixing these alternative conceptions in pupils' understanding. At worst, the incorrect understanding of how key computing constructs operate is not corrected, hampering effective progress for pupils. Given this analysis, it is easy to understand the common view among educators that computational ability is innate, rather than that there are developmental stages of which the educators are simply unaware.
- **Problem solving is introduced too early**, before novices have had an opportunity to develop even the beginnings of the pattern-based problem solving library that any expert will have developed. At such an early stage,

[2] http://www.sqa.org.uk/sqa/48477.html

problem solving can only take place from first principles, a form of problem solving that is incredibly heavy on the human cognitive system. Such a cognitive overload leaves no room for any real learning, explaining why researchers have found that novices are unable to remember the details of the difficulties they faced when in problem solving laboratory sessions; they could only remember the associated emotions of pain or joy. Instead of early problem solving, a form of so-called *cognitive apprenticeship* is required, where novices can follow the cognitive processes used by an expert to solve a problem. The use of *worked examples* is one approach to modeling such expert behaviour, and by studying a large number of worked examples, a novice should be able to develop the foundations of the expert's pattern library at a fraction of the cost and pain of more traditional approaches.

Recognising these two prevalent failings, we have developed a series of approaches to teaching, thereby operationalising the research findings in a way that is relatively easy to adopt in the classroom. We build on research that demonstrates a near equivalence between code explaining and code writing skills – if you can explain its behaviour, you can write it – but that code reading must come before either of these. Our approaches are summarised as follows:

1. Appreciating the scale of the task. This is a component for teachers alone, not necessarily to be used in classes, to let them see how much knowledge and skill we are typically expecting a novice to acquire in only a few weeks of classes.
2. Program comprehension 1. We use Schulte's *Block Model* of program comprehension as a corner stone of the programme. As a starting point, we introduce simple exercises to help novices learn to name programming concepts and describe their purpose and operation. This is working at the lowest level of Shulte's model.
3. Program comprehension 2. We have developed a series of paper-and-pencil exercises that use code tracing as a learning activity. Typical tracing, as used by experts to track variable values, is much shunned by novices. A cognitive load analysis explains why, since much that is implicitly understood by the expert, such as the static control flow, expression evaluation and statement execution rules must be consciously considered by a novice. Our exercises enable much of this understanding to be explicitly expressed on the tracing sheet, freeing up the novice's brain sufficiently to make the activity achievable. The sheets present a precise mental model of how constructs are executed, and can be used to both present the right models and test to ensure that students have them. We have fully developed exercises for expression evaluation, simple store and control flow semantics, and draft exercises for subprogram execution and complex data structures.
4. Common mistakes and alternative conceptions. Research shows that knowledge of typical learner misconceptions is more important than core subject knowledge for successful teaching. We have brought together a long list of misconceptions drawn from a range of publications.
5. Peer Instruction. This is a much-researched general pedagogical technique from the *flipped classroom* stable of instructional designs, regularly producing 'times 2' gains in learning over traditional techniques. The value of this approach is that: students practice both using the language of the subject and hearing it being used; their peers' description of their understanding, they having maybe only just mastered the concepts themselves, are more appropriate to the listeners' stage of development than their teacher who learned years ago; and the requirement to justify their answers to questions forces them to dig deep into their own conceptual understanding.

6. Introduction to worked examples. This covers the issues with the typical problem-solving-first approach to learning computing skills, and offers worked examples as a more efficient learning approach. Teachers are introduced to best practices in worked example creation and also an on-line worked examples system developed alongside the PLAN C project that they can use with their pupils.
7. Worked examples and subgoal labeling. Recent research is showing that the steps of procedures can be learned significantly faster if each step is, or coherent groups of steps are given an appropriate *label*. This named chunking of the overall procedure improves retention and recall.
8. New Practice Surgery. This is one example of a number of techniques we introduced for sharing so-called *pedagogical content knowledge*. This is the gold-dust of teaching and learning – the wisdom that separates the expert teacher from the novice. The New Practice Surgery recognises that adopting new teaching practice is challenging, even scary, and if it doesn't go well in front of the class, there is little willingness to try again. The Surgery session gives an opportunity for teachers in groups of 3 or 4 to share their experience of using new approaches in a structured fashion, such that the positive can be adopted, and the challenging aspects discussed and improved, drawing on the wisdom of all present.

It should be noted that this is not typical computing professional development that in the past has mainly consisted of the teachers' learning new languages and systems. Instead, these approaches involve developing key conceptual understandings about the subject, learning about improved general and subject-specific pedagogical techniques and developing a practice of effective sharing of wisdom between teachers.

Lead Teacher Training

We needed leaders all around the country to set up localised teacher professional development hubs as recommended by the Donaldson report. These are the PLAN C *lead teachers*.

We have trained 55 lead teachers from all over Scotland, in three cohorts. The training of each cohort took place over 4 sessions – a 1.5 day opener, then two single days, and then a 1.5 day closer. Each session was separated by 2-3 weeks. This format aimed to give the lead teachers a sound experience of multiple professional development cycles – introduction to concepts and practices in one session, time to try it out in between sessions, reporting back and reflecting on experiences with other teachers in the next session. This cyclical nature is recognised to be a vastly superior form of professional development that the more typical big-bang 'go on a one-day course' style.

The overnight nature of the opening and closing sessions was designed to ensure a significant social as well as professional experience, helping to form a strong national network of teachers – but also to model to the lead teachers the kind of social experience we hoped they would foster later on in their local groups.

The sessions were all led jointly by the two project officers to model how much more relaxing team teaching is. We aimed to act more as facilitators of the teachers' learning rather than instructors. We introduced concepts and professional development activities, had the teachers work through the activities themselves, and then led a discussion about the experience. We hoped to model exactly how we wanted them to be with their own groups.

Local hub programme

We have supported the lead teachers to create 25 local teacher hubs spanning the country, mostly led by a pair of leads, occasionally by just one, or by three. A significant publicity campaign involved direct contact from Scottish Government to the Directors of Education of all 32 Local Authorities to encourage teacher uptake of the programme at a local level. Education Scotland included details in their weekly update bulletins to all schools. We mailed the computing department in every school in Scotland with details of the programme. Around 350 of the 650 CS teachers initially signed up on-line to take part. Hubs meet approximately monthly and are led by the lead teachers following materials mainly based on their own lead teacher training, but enhanced by the project officers. Lead teachers have a budget to organise the venue and arrange for any catering, and are paid a small amount for their time in preparation and presentation. If working in pairs or threes, they usually meet up in advance of each session to ensure everyone is thoroughly prepared.

The project officers have kept in touch with the lead teachers, initially by arranging to call them between sessions to brief/debrief on sessions about to be, or just, run. Latterly, these communications have been on a more ad-hoc basis, but the hubs are still operating well.

A Moodle virtual learning environment has been set up to host all materials. There is a specific Moodle 'course' for each local hub, enabling the lead teachers of that hub to share the particular materials they used and to allow teachers to communicate via forum. And also a lead teacher 'course' where the original versions of all materials created by the project officers is hosted.

Additional activities

We have run, or plan to run, a number of additional activities:

- *Launch Events*. At the start of the programme, we ran three 1-day events around the country to provide immediate input in areas identified by teachers in a poll we ran as the most problematic. Around 150 teachers attended in total.
- *Craft the Curriculum* events. Collaboratively with Education Scotland, we have assembled industry staff, academics and teachers for two 2-day workshops to develop relevant, innovative teaching materials for the more advanced learning outcomes of the new Scottish qualifications, so far developing 8 activities.
- *Unlocking the Thinking behind Computing* on-line course. This very basic programming course uses programming *as a vehicle only* to develop computational thinking skills in technology-averse primary teachers. It uses cutting-edge approaches to on-line education, such as on-line discussion sessions between pairs of participants to emulate Peer Instruction.
- We ran a special session for the question-setting teams of the Scottish Qualifications Authority to deepen understanding of the issues around code comprehension and how to assess pupils' ability to understand code.
- We are key contributors to early plans for a new Scottish computing curriculum starting at the primary level.
- We will support the lead teachers to gain professional recognition for their work from the General Teaching Council Scotland (GTCS).
- We have gained Google funding in their CS4HS programme to make all the materials developed to date available internationally in a self-study format. This is essential locally as well, since it is clear that through no fault of their own, not all teachers can physically attend the local hub meetings. Some have child-care issues, some are in very geographically-dispersed areas.
- For teachers in outlying regions, we are exploring using virtual meeting techniques: one local hub in Perth is successfully using a local university's video-conferencing suite to bring in teachers from the Orkney Islands.

2. Evidence of availability of the curricula materials

A quick link to the section of our Moodle site for teachers is located at:

http://bit.ly/plancmaterials

A special account and password has been set up to enable viewing of the materials. Username: *specialplancuser* and Password: *planc2015*. This will give you access to the so-called *Lead Teacher BACKROOM* course, which contains all materials. It is these materials that are currently available to local teachers in the programme, depending on how far their local hub has progressed through the set of materials.

Please do not circulate this document widely with this username and password included. It has only been included for evaluation of this submission. Confidential attendance information is also accessible from this page. The materials here are made available by lead teachers to any local teachers taking part in the programme, by copying them to another part of the site. As they stand, the materials would not be ideal for teachers not taking part in the local hub programme, but as noted above, in the next 3-4 months, we will be making self-study versions of all these materials.

3. Evidence of Impact

Two sources of evidence are used to show evidence of impact.

First, an independent evaluation of the project was a part of the original proposal. This has been carried out by a former computing teacher in Scottish schools from around 10-15 years ago, since then a leading player in the Scottish education system for a number of years and now an independent consultant. The evaluator has attended sessions of the vast majority of the local hubs, many on several occasions, and so has been able to develop a strong picture of how the project as a whole is operating.

Second, lead teachers were polled recently on their views of the project, as part of another interaction with them.

Quantitative measurement of the adoption of the materials in school classrooms is about to take place, but is unfortunately not available in time for this submission.

From the project's independent evaluation report

The following excerpts are drawn from the a draft of the independent evaluation of the project, each addressing a specific aim of the project, as identified by the evaluator.

Elaboration and development of computational thinking as a more substantial and coherent approach to the teaching of computing science.

This aspect of PLAN C sets it apart from most other teacher professional development initiatives in that it seeks to redefine the nature of CS and consequently how best to teach it. Rather than subject content being to the fore it is the intellectual skills that need to be developed which are being highlighted. These intellectual skills have always been present in CS but have too often been hidden and left to each teacher to try and work out with varying degrees of success.

Computational thinking is a set of core or essential intellectual skills for CS that once acquired enable any technology to be understood and applied. Through the lens of computational thinking an education in computing science is not about any particular technology, it is about developing the intellectual skills that are embodied by a computer scientist. The technologies studied whilst being important and interesting are at one level just vehicles for developing those higher-order cognitive skills.

Evidence that this idea has been taken on by the lead teachers and local hub participants is clear from the interviews held with PLAN C teachers. Early indications are that this shift *towards computational thinking as the basis of CS has helped to renew confidence in the subject discipline and provide teachers with a clear rationale why computing science should be an important aspect of every child's formal curriculum.*

The creation of a network of lead teachers among computing science teachers

This aspect of the project has been highly successful. A group of 55 lead teachers has been through an intensive course and 50 of these have in turn activated and managed a total of 25 local hubs.

Three initial lead teachers courses were held in Glasgow, Edinburgh and Dundee during school session 2013/14. There was also a 'fast track' additional accelerated one-day course held in November 2014 with a follow-up day in March 2015. This 'fast track' course was arranged to help fill some of the gaps in the local hub coverage. All four of the lead teacher courses were judged to be of an exceptionally high quality by those taking part. The lead teachers appeared to relish the opportunity to grapple with a challenging course that took them well out of their comfort zones. Feedback suggests that they were a largely self-selected group of volunteers crying out for exactly the kind of high-quality, subject specific professional learning opportunities that PLAN C offered.

At the time of writing almost all of the lead teachers surveyed had used at least one thing they had learned through PLAN C with their classes. The most frequently used part of the course was the peer instruction questioning technique, either for revision purposes or to help uncover common misconceptions/alternative conceptions of key concepts. The second most used aspect of the course was the code comprehension through tracing methodology – TRACS. Despite it being almost a year on from three out of the four lead teacher courses it is clear that it will take some time before PLAN C is fully integrated into the classroom practice of the lead teachers. This in turn suggests at least another year before the full impact will be realised in the classroom practice of the local hub participants.

Creation of a number of sustainable local hubs that bring together lead teachers and other computing science teachers to share effective classroom practice.

As noted above there are currently 25 local hubs established with 50 lead teachers actively involved. By February 2015 a total of 305 computing teachers had attended at least one local hub meeting with 198 of these attending at least 50% of possible sessions. This means that 360 CS teachers in Scotland have had direct contact with PLAN C through either lead teacher training or attendance at a local hub (305 plus the 55 original lead teachers).

The varying contexts of the hubs mean that there is no single approach that could meet all local needs. For example the two Fife based local hubs launched on an in- service day at the start of the 2014/15 session and all of the 55 computing teachers from the local authority were able to attend. Edinburgh and the Lothians based hubs take advantage of the asymmetric week and meet on those Friday afternoons scheduled for subject/departmental time. Whilst others, such as the hub covering Clackmannanshire and Stirling, tend to meet after school to accommodate travel time for teachers across local authority boundaries. In each of these cases the lead teachers have worked with their local computing teacher peers to develop a model that works for their particular circumstances.

Initially there were 27 hubs planned and at the time of writing three of these have either never met or are currently inactive. These are: Argyll and Bute; Angus; and Inverclyde. The reasons for this range from lead teachers going on maternity leave (Angus), difficulties at local authority level (Inverclyde) and logistical issues (Argyll and Bute).

From observations and participant feedback the local hubs have been highly successful and well received by the vast majority of participants. With 75% of participants surveyed indicating that they had started to use aspects of what they had learned through PLAN C with their classes and almost all expecting to use more over the next year once they had time to integrate the ideas into their courses.

Collection of a body of evidence on what works with respect to computing science teaching and why it works, with a view to impacting on policy/practice at national levels

This aspect of the project has also been successful with Quintin and Peter drawing on academic research studies relating to CS pedagogy. The BRACElet project, for example, is research from Australia around novice programmers and their need to be able to identify particular computational concepts in code as well as trace and explain it before they can successfully write code of their own. The main findings of this research have been summarised and packaged in a form that was accessible for the PLAN C teachers, many of whom have already started the process of implementing a change from a 'code creation orientation' to a 'code comprehension orientation' in their classroom practice. Similarly the methodology of peer instruction has been used as a means of uncovering conceptual misunderstandings/alternative conceptions in CS for PLAN C teachers. This highly effective approach draws directly from Eric Mazur's research into undergraduate physics teaching at Harvard. PLAN C has also drawn on Richard Catrambones work on improving worked examples through sub- goal labeling amongst many).

In relation to impacting on national policy Quintin and Peter have been deeply involved in building relationships with key stakeholders in Scottish education and beyond to significant effect. Their work alongside others including Kate Farrell and Computing at School Scotland (CAS Scotland) has created a voice for CS in schools that has long been absent. At a national policy level these efforts have generated the essential connective tissue between otherwise disparate groups: initial teacher education; Education Scotland; to some extent the Scottish Qualifications Authority (SQA); the higher education CS community (SICSA) and the schoolteachers themselves. The importance of such genuine valuable communication and partnership working should not be underestimated.

Teachers: "What are your views on the PLAN C project?"

The following quotes come from a question titled "What are your views of the PLAN C project?" to teachers involved:

"I really like the PLAN C project and what it is setting out to achieve. The PLAN C project has made me think a lot about teaching 'the hidden machine' - although this was something that I have always worked hard to include in my own lessons. At times, I've had difficulty trying to persuade more reluctant members of my department that it is simply not good enough to assume that (1) programming is hard (2) programming cannot be taught and (3) some pupils simply cannot program. PLAN C confirmed what I had always believed: if the content is presented in ways that are accessible, then all pupils can achieve. And more importantly, they can understand how the 'black box' works."

"It's been a brilliant opportunity to discuss practice with colleagues and to deepen my own understanding of the specific pedagogical challenges that face computing teachers. Our hub has been really well attended and we've had a lot of thoughtful and lively debate about the material we've been presenting. It's been a great experience to get (and to be able to disseminate) subject specific CPD that is centred on teaching rather than content."

"Plan C is a unique and exciting initiative that promises to shape the development and delivery of Computing Science in schools in ways that are underpinned by research findings and by practitioner experience. A major action research initiative that may well mark a turning point in the teaching of the subject in schools, Further Education and Higher Education."

"The PLAN C project has highlighted the need to evolve the way we teach computing to students. This is highlighted in the way that I now teach programming. PLAN C has also given me the opportunity to regularly hold meetings with colleagues in the Borders, which is something that hasn't happen very often in the past. Finally this project has brought the

need for schools to have computing departments to the attention to Headteachers and regional educational departments."

"Refreshing, unique and, ultimately, overdue. Too often 'traditional' lesson styles, content and pedagogy are used because 'that's the way it's always been done' and as a subject area we need to re-evaluate these for our current CS pupils. Not only to deepen their understanding in the subject area and build a foundation for future study but, importantly, to engage and enthuse pupils of all skills, ability and motivation. I think PLAN C is thought provoking and a major step in the right direction for teaching (and learning) our subject."

Teachers: How has PLANC helped?

'Spending dedicated time with fellow computing teachers working on developing my own understanding and skill set rather than delivering in-service functions as is most commonly the case for an IT pro.'

'Discussion and reflection on my own practice and that of others.' 'Colleague discussion, similarity of problems encountered.' 'Sharing best practice and new ideas.' 'Ideas from other colleagues to try.'

'Exemplification of different methods.' 'Getting the chance to try it out myself and discuss ideas with others.'

'Examples given and discussion time with a larger group of staff than just in the department.'

'Example questions supported – plus the experience of discussing format and application with colleagues. This has improved my own questioning techniques.'

'Examples provided by the course with a chance to implement while fresh in the head.'

'[Peer instruction] questions used in class for a range of topics challenging students to think about understanding.'

'[code comprehension techniques such as] Block model and TRACS are a focus of future development.'

'Sharing of ideas and questions – great resource for PI.'

'The opportunity to talk the job and engage in discussions re pedagogy and approaches/ideas is invaluable. It is a real professionalisation of review and development time. The chance to evaluate materials and trial materials in a supportive environment is of value.'

4. Letters of Support

There follow two letters of support:

- From Professor Alan Bundy FRS, of the University of Edinburgh, formerly vice-president of BCS, The Chartered Institute for IT, and more recently their representative on a series of Scottish initiatives to improve computing science teaching in schools – the Learned Societies Group on Science Education, the Royal Society of Edinburgh exemplification project for computing science and on the steering group for the PLAN C project.
- From Laurie O'Donnell, the Scottish Government's independent evaluator of the PLAN C project, a former school teacher of computing, a national education leader in Scotland, and now an independent consultant.

School of informatics

THE UNIVERSITY OF EDINBURGH

June 2, 2015

CENTRE *for* INTELLIGENT SYSTEMS *and their* APPLICATIONS

SCHOOL *of* INFORMATICS
The University of Edinburgh
The Informatics Forum
10 Crichton Street
Edinburgh EH8 9AB

Telephone 0131 650 2716
Secretary 0131 650 2733
Fax 0131 650 6513
E-mail A.Bundy@ed.ac.uk
URL http://homepages.inf.ed.ac.uk/bundy/

Informatics Europe
2015 Best Practices in Education Award

To whom it may concern,

Letter of Support for PLAN C Submission

It is a pleasure to write in support of the submission of the Professional Learning Network in Computing (PLAN C) for the 2015 Best Practices in Education Award.

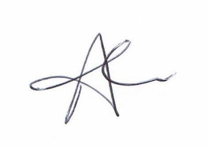
PLAN C has revolutionised the provision of continuous professional development for Computing school teachers. It has used the latest, evidence-based pedagogy to empower teachers to provide the kind of Computing education that the World needs in the 21^{st} century. It combines the teaching of programming with the ability to think computationally to increase student's problem solving abilities. It uses regional hubs, led by PLAN C trained lead teachers, to engage with a majority of Computing teachers in Scotland. Active learning by these teachers has them apply and develop pedagogic research, which increases their confidence and engagement. This engagement will also help the CPD scheme become self-sustaining after the pump-priming Government funding terminates.

PLAN C comes at a critical time in Scottish Computing teaching. An international revolution in Computing teaching is underway, from office skills to programming and computational thinking. The new Scottish Curriculum for Excellence has accelerated this revolution in Scotland. While this is a very welcome development, it has taken the teachers well outside their comfort zone and made them desperate for CPD to capacitate them for their new role. The Scottish Government has recognised this need and funded the PLAN C programme at £400k over two years. PLAN C is being independent evaluated and has exceeded its objectives in terms of coverage and the enthusiasm of the teachers.

But this is not just relevant to Scotland. The same Computing teacher revolution is now happening World wide. Scotland is a laboratory for the development of new teaching materials and CPD. Our experience will be invaluable to our colleagues elsewhere.

The PLAN C submission to the 2015 Best Practices in Education Award has my strongest support. Please do not hesitate to contact me if I can provide further information.

Yours sincerely,



Professor Alan Bundy

Laurie O'Donnell Ltd

3 June 2015

To Whom It May Concern:

Informatics Europe 2015 Best Practices in Education Award:
Letter of Support for the PLAN C Project

I am happy to confirm that I carried out an independent evaluation of the PLAN C project for the Scottish Government in 2014/15.

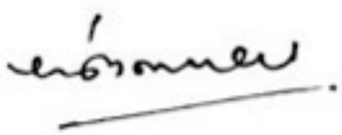
The excerpts quoted from my evaluation by Professor Quintin Cutts and Peter Donaldson in section 3 of their submission for the above award accurately represent my findings.

PLAN C is a highly ambitious and innovative professional development project that has been implemented in an exemplary fashion by Quintin and Peter. There is already a considerable body of evidence to support the assertion that PLAN C is making a major impact on how CS teachers in Scotland think about their subject and its related professional practice.

I have attended over 40 PLAN C local hub meetings across the country and have been impressed by the level of engagement from the participating teachers. The quality of the professional conversations I have witnessed has been of an exceptional standard as teachers grapple with the challenge of applying research on CS specific pedagogy in their classrooms.

Overall PLAN C is already making a positive contribution to reinvigorating CS in Scottish schools and may also be helping to elaborate a collegiate, research informed model for professional learning across the curriculum.

Yours faithfully



Laurie O'Donnell
Director

Laurie O'Donnell Ltd
E: laurie@laurieodonnell.co.uk
T: +44(0)790 999 4672
W: www.laurieodonnell.co.uk

5. Reference list

Below is a list of key papers that influenced the design and structure of the PLAN C programme and the individual techniques and activities.

Overall Design - General

Thomas R. Guskey (2002) Professional Development and Teacher Change, Teachers and Teaching: theory and practice, 8:3, 381-391

Richardson, V. (1990). Significant and worthwhile change in teaching practice. Educational researcher, 19(7), 10-18.

Overall Design - CS Specific

Briana B. Morrison , Lijun Ni , Mark Guzdial (2012), Adapting the disciplinary commons model for high school teachers: improving recruitment, creating community, Proceedings of the ninth annual international conference on International Computing Education Research, September 09-11, 2012, Auckland, New Zealand

Fincher, S., & Tenenberg, J. (2006). Using Theory to Inform Capacity- Building: Bootstrapping Communities of Practice in Computer Science Education Research. Journal of Engineering Education, 95(4), 265-277.

Kolikant, Y. B. D., & Pollack, S. (2004, June). Community-oriented pedagogy for in-service CS teacher training. In ACM SIGCSE Bulletin (Vol. 36, No. 3, pp. 191-195). ACM.

Activity	Associated Research
Activity 1: Introduction to PLAN C and Cup of Tea Exercise	CS Specific Briana B. Morrison , Lijun Ni , Mark Guzdial, Adapting the disciplinary commons model for high school teachers: improving recruitment, creating community, Proceedings of the ninth annual international conference on International Computing Education Research, September 09-11, 2012, Auckland, New Zealand Buchholz, M., Saeli, M., & Schulte, C. (2013, November). PCK and reflection in computer science teacher education. In Proceedings of the 8th Workshop in Primary and Secondary Computing Education (pp. 8-16). ACM.
Activity 2: Program visualisation and TRACS and Activity 3: TRACS review	CS Specific Juha Sorva (2013), Notional machines and introductory programming education, ACM Transactions on Computing Education (TOCE), v.13 n.2, p.1-31, June 2013 Matthew Hertz, Maria Jump (2013), Trace-Based Teaching in Early Programming Courses, SIGCSE'13, March 6–9, 2013, Denver, Colorado, USA. Anne Venables , Grace Tan , Raymond Lister (2009), A closer look at tracing, explaining and code writing skills in the novice programmer, Proceedings of the fifth international workshop on Computing education research workshop, August 10-11, 2009, Berkeley, CA, USA Mike Lopez, Jacqueline Whalley, Phil Robbins and Raymond Lister (2008), Relationships Between Reading, Tracing and Writing Skills in Introductory Programming, ICER'08, September 6–7, 2008, Sydney, Australia. Soloway, E. (1986). Learning to program= learning to construct

	mechanisms and explanations. Communications of the ACM, 29(9), 850-858. du Boulay, B. (1986). Some Difficulties of Learning to Program. Journal of Educational Computing Research, 2(1):57–73.
Activity 4: Mistakes, alternative conceptions and mental models	General Sadler P, Sonnert G, Coyle H, Cook-Smith N, Miller J. (2013) The influence of teachers' knowledge on student learning in middle school physical science classrooms. Am Educ Res J 2013;50:1020-1049. CS Specific Identifying student misconceptions of programming Teemu Sirkiä , Juha Sorva (2012), Exploring programming misconceptions: an analysis of student mistakes in visual program simulation exercises, Proceedings of the 12th Koli Calling International Conference on Computing Education Research, p.19-28, November 15-18, 2012, Koli, Finland Clancy, M. (2004). Misconceptions and attitudes that interfere with learning to program. Computer science education research, 85-100. Soloway, E. (1986). Learning to program= learning to construct mechanisms and explanations. Communications of the ACM, 29(9), 850-858. Pea, R. D. (1986). Language-independent conceptual" bugs" in novice programming. Journal of Educational Computing Research, 2(1), 25-36. Jeffrey Bonar , Elliot Soloway (1985), Preprogramming knowledge: a major source of misconceptions in novice programmers, Human-Computer Interaction, v.1 n.2, p.133-161, June 1985
Activity 5: The Why and How of Haggis	CS Specific Quintin Cutts, Richard Connor, Greg Michaelson, Peter Donaldson (2014), Code or (not code): separating formal and natural language in CS education, Proceedings of the 9th Workshop in Primary and Secondary Computing Education, p20-28 Allison Elliott Tew , Mark Guzdial (2011), The FCS1: a language independent assessment of CS1 knowledge, Proceedings of the 42nd ACM technical symposium on Computer science education, March 09-12, 2011, Dallas, TX, USA
Activity 6: Peer Instruction	General Catherine H. Crouch and Eric Mazur (2001), Peer Instruction: Ten Years of Experience and Results, Am. J. Phys., 69, 970-977 CS Specific Simon, B. and Cutts, Q. (2012) Peer Instruction: A Teaching Method to Foster Deep Understanding. Comm. ACM 55(2), pp 27-29. Simon, B., Parris, J., & Spacco, J. (2013, March). How we teach impacts student learning: peer instruction vs. lecture in CS0. In Proceeding of the 44th ACM technical symposium on Computer science education (pp. 41-46). ACM. Cutts, Q., Esper, S., Fecho, M., Foster, S. R., & Simon, B. (2012, September). The abstraction transition taxonomy: developing desired learning outcomes through the lens of situated cognition. In Proceedings of the ninth annual international conference on International computing

	education research (pp. 63-70). ACM. Simon, B., & Cutts, Q. (2012). How to implement a peer instruction-designed CS principles course. ACM Inroads, 3(2), 72-74. Porter, L., Bailey Lee, C., Simon, B., & Zingaro, D. (2011, August). Peer instruction: do students really learn from peer discussion in computing?. In Proceedings of the seventh international workshop on Computing education research (pp. 45-52). ACM.
Activity 7: Understanding Code Comprehension- Intro to the Block Model and Activity 8: Code Comprehension for HTML and CSS	CS Specific Busjahn, T., & Schulte, C. (2013, November). The use of code reading in teaching programming. In Proceedings of the 13th Koli Calling International Conference on Computing Education Research (pp. 3-11). ACM. Schulte, C. (2008, September). Block Model: an educational model of program comprehension as a tool for a scholarly approach to teaching. In Proceedings of the Fourth international Workshop on Computing Education Research (pp. 149-160). ACM. Mike Lopez, Jacqueline Whalley, Phil Robbins and Raymond Lister (2008), Relationships Between Reading, Tracing and Writing Skills in Introductory Programming, ICER'08, September 6–7, 2008, Sydney, Australia. Anne Venables , Grace Tan , Raymond Lister (2009), A closer look at tracing, explaining and code writing skills in the novice programmer, Proceedings of the fifth international workshop on Computing education research workshop, August 10-11, 2009, Berkeley, CA, USA
Activity 9: Impact Of PLAN C So Far	General Allen, R. M., & Casbergue, R. M. (1997). Evolution of novice through expert teachers' recall: Implications for effective reflection on practice. Teaching and Teacher Education, 13(7), 741-755. CS Specific Buchholz, M., Saeli, M., & Schulte, C. (2013, November). PCK and reflection in computer science teacher education. In Proceedings of the 8th Workshop in Primary and Secondary Computing Education (pp. 8-16). ACM.
Activity 10: Introduction to Worked Examples and WEAVE the online worked example viewer	General Schwonke, Rolf, et al. (2009) "The worked-example effect: Not an artefact of lousy control conditions." Computers in Human Behavior 25.2: p258-266. Sweller, J. (2006). The worked example effect and human cognition. Learning and Instruction, 16(2) p165–169 Paas, F., Renkl, A., & Sweller, J. (2003). Cognitive load theory and instructional design: Recent developments. Educational psychologist, 38(1), p1-4 CS Specific Song, Yulun (2015) An authoring and presentation environment for interactive worked examples, University of Glasgow Phd Thesis. Ben Skudder, Andrew Luxton-Reilly (2014), Worked examples in computer science, ACE '14 Proceedings of the Sixteenth Australasian Computing Education Conference - Volume 148 Pages 59-64

	Kinnunen, P., & Simon, B. (2010, August). Experiencing programming assignments in CS1: the emotional toll. In Proceedings of the Sixth international workshop on Computing education research (pp. 77-86). ACM.
Activity 11: Improving Worked Examples- Subgoal labelling	General Catrambone, R. (1998). The subgoal learning model: Creating better examples so that students can solve novel problems. Journal of Experimental Psychology: General, 127(4), 355. CS Specific Margulieux, L. E., Catrambone, R., & Guzdial, M. (2013) Subgoal Labeled Worked Examples Improve K-12 Teacher Performance in Computer Programming Training Margulieux, L. E., Guzdial, M., & Catrambone, R. (September 2012). "Subgoal-labeled instructional material improves performance and transfer in learning to develop mobile applications". ICER '12, Proceedings of the ninth annual international conference on International computing education research: 71–78.
Activity 12: Contribution of a Computing Education- The UV Model	CS Specific Cutts, Q., Esper, S., Fecho, M., Foster, S. R., & Simon, B. (2012, September). The abstraction transition taxonomy: developing desired learning outcomes through the lens of situated cognition. In Proceedings of the ninth annual international conference on International computing education research (pp. 63-70). ACM. Soloway, E. (1986). Learning to program= learning to construct mechanisms and explanations. Communications of the ACM, 29(9), 850-858.
Activity 13: New Practice Surgery	General Richardson, V. (1998). How teachers change. Focus on basics, 2(C), 1-10. Allen, R. M., & Casbergue, R. M. (1997). Evolution of novice through expert teachers' recall: Implications for effective reflection on practice. Teaching and Teacher Education, 13(7), 741-755. Day, C. (1993). Reflection: a necessary but not sufficient condition for professional development. British educational research journal, 19(1), 83-93. CS Specific Buchholz, M., Saeli, M., & Schulte, C. (2013, November). PCK and reflection in computer science teacher education. In Proceedings of the 8th Workshop in Primary and Secondary Computing Education (pp. 8-16). ACM.
Activity 14: Pattern Orientated Instruction	CS Specific Ginat, D., Shifroni, E., & Menashe, E. (2011). Transfer, cognitive load, and program design difficulties. In Informatics in Schools. Contributing to 21st Century Education (pp. 165-176). Springer Berlin Heidelberg. Haberman, B., Muller, O., & Averbuch, H. (2008). Multi-facet Problem Comprehension: Utilizing an Algorithmic Idea in Different Contexts. In Informatics Education-Supporting Computational Thinking (pp. 180-191). Springer Berlin Heidelberg. Muller, O., Ginat, D., & Haberman, B. (2007, June). Pattern-oriented instruction and its influence on problem decomposition and solution construction. In ACM SIGCSE Bulletin (Vol. 39, No. 3, pp. 151-155). ACM.

	Muller, O. (2005, October). Pattern oriented instruction and the enhancement of analogical reasoning. In Proceedings of the first international workshop on Computing education research (pp. 57-67). ACM. Clancy, M. J., & Linn, M. C. (1999, March). Patterns and pedagogy. In ACM SIGCSE Bulletin (Vol. 31, No. 1, pp. 37-42). ACM.
Activity 15: Cooperative Code Review	CS Specific Christopher D. Hundhausen, Anukrati Agrawal and Pawan Agarwal (2013), Talking about Code: Integrating Pedagogical Code Reviews into Early Computing Courses, ACM Transactions on Computing Education (TOCE) - Special Issue on Alternatives to Lecture in the Computer Science Classroom TOCE Volume 13 Issue 3, August 2013, Article No. 14 Vennila Ramalingam , Deborah LaBelle , Susan Wiedenbeck (2004), Self-efficacy and mental models in learning to program, ACM SIGCSE Bulletin, v.36 n.3, September 2004
Activity 16: Variable Roles	CS Specific Sajaniemi, J., & Kuittinen, M. (2005). An experiment on using roles of variables in teaching introductory programming. Computer Science Education, 15(1), 59-82. Laakso M.-J., Malmi L., Korhonen A., Rajala T., Kaila E. Salakoski T. (2008) Using Roles of Variables to Enhance Novice's Debugging Work. Issues in Informing Science and Information Technology, Volume 5, 281-295. Sorva J., Karavirta V., Korhonen A. (2007) Roles of Variables in Teaching. Journal of Information Technology Education, 6, 407-423. (Also to be presented in Informing Science + IT Education Joint Conference (InSITE 2007), Ljubljana, Slovenia, June 2007.) Sorva J. (2007) A Roles-Based Approach to Variable-Oriented Programming. J. Sajaniemi, M. Tukiainen, R. Bednarik, S. Nevalainen (eds.) Proceedings of the 19th Annual Workshop of the Psychology of Programming Interest Group (PPIG07), Joensuu, Finland, July, 2007. International Proceedings Series 7, University of Joensuu, Department of Computer Science and Statistics, 116-128
Activity 17: Illustrating the difference between values & references	CS Specific Juha Sorva (2013), Notional machines and introductory programming education, ACM Transactions on Computing Education (TOCE), v.13 n.2, p.1-31, June 2013
Activity 18: Beliefs about learning CS	General Kagan, Dona M. (1992) Implication of research on teacher belief. Educational psychologist 27.1: 65-90. CS Specific Cutts, Q. , Cutts, E., Draper, S. , O'Donnell, P. , and Saffrey, P. (2010) Manipulating mindset to positively influence introductory programming performance. In: SIGCSE '10 Proceedings of the 41st ACM SIGCSE Symposium, 10-13 Mar 2010, Milwaukee, USA. Murphy, L., & Thomas, L. (2008). Dangers of a fixed mindset: implications of self-theories research for computer science education. ACM SIGCSE Bulletin, 40(3), 271-275.