

Targeted remediation for a computer programming course using student facilitators

Steve Draper^{*}
Department of Psychology
University of Glasgow
s.draper@psy.gla.ac.uk
Tel: 0141 330 5089

Quintin Cutts
Department of Computing Science
University of Glasgow
Quintin@dcs.gla.ac.uk
Tel: 0141 330 5619

Abstract

The first results from a new intervention are presented. This paper describes the originally identified need (a very high failure rate in a subgroup completing a computer programming course, with important consequences for student dropout); a remediation intervention (timetabled student study groups facilitated by senior students, not staff); the positive effect on subsequent exam measures; and some strongly positive qualitative feedback from the students. Finally some educational analysis is presented: both the original rationale for the intervention, and a reinterpretation in the light of the outcomes.

Keywords: retention, computer programming, peer tutoring, remediation.

The problem identified

A few years ago, numbers of applicants for computing science courses collapsed generally at least across the UK. At the university in this study, there were 480 in the first year class in 2001, but by 2005 only about 180. From 2002 the progression requirement from year 1 to year 2 was lowered from the original firm requirement of two

^{*} Corresponding author

C grades, the more important of the two being the grade obtained in the first year programming course. In summer 2005, an analysis was done of what had happened to the students who had entered year 2 but who would not have previously been allowed to. It found that on average 89% of these had failed to get a C grade in the year 2 computing course, and so could not progress to year 3, and many of these had subsequently dropped out of the university.

The context of this is a university with four year degrees, and where students do several different subjects in the first two years before specialising in the third and fourth years. While performing poorly in one discipline (computing science) does not necessarily prevent progression to a different subject in later years, in practice it often does. Within computing science, it is computer programming that is most often the key topic leading to failure.

Dropout statistics are fraught with complexities, especially in the short term, since it is only failure to complete a degree after say ten years that is definitive from all viewpoints. Here we will say that the only fully satisfactory outcome of taking a programming course is to achieve a grade C. Those who do not attain this, we classify here, for the purposes of the new accelerator course, as "at-risk". In addition, it is desirable that they achieve a grade C average across all their computing courses which would definitely qualify them for progression to computing science courses in the next year. Using two years of figures, we can say that of 51 students who were at-risk on entry to the year 2 course, 27 had already dropped out of the university when the analysis was done and a further 19 were repeating a year. In the past a large majority of these fail to finish a degree. We can therefore say that over half of those "at-risk" are already known to have become dropouts, in reality about four fifths probably will.

In the three years preceding the intervention, of those entering the year two computing course at risk, only a very small proportion improved up out of the at-risk category to gain a grade C average to guarantee progression: 7%, 16%, and 10% respectively. The narrow aim of the intervention was to have them achieve at least a grade C in the year two programming course, and the wider aim was thereby to reduce the number who would subsequently be lost to the university.

The remedy designed

The remedy that was designed and implemented was an "accelerator course": an additional mini-course for programming run in the first few weeks of the second year, primarily for the at-risk students. It consisted of sessions run by final year students who had experience of facilitating Peer Assisted Learning sessions. Whilst aimed at this target set of students who were strongly urged to attend, it was also open to other students. There were (for each student) an initial two hour session run by a member of staff, then five sessions per week for the first two weeks of term, then three per week for the next two weeks, then two per week for another two weeks. That concluded the originally planned set, although student demand led to continuing sessions for the remainder of the term. The first session began the mini-course with motivation and reflection conducted by a staff member. After that the pattern was working on programming problems individually, and when problems arose peer discussion was first used to attempt resolution, followed by expert input from the facilitators if the whole group was stuck on the same issue. A key aim was the establishment of steady and purposeful work habits, and hence increased time on task, but even more important (we think in retrospect) was the guarantee of expert knowledge if required so that students were not left stuck. This is discussed below in the section on "educational interpretation".

The institutional financial perspective

From the university's point of view, it loses about £7,000 per year for each student who drops out before completing a degree leaving vacancies in courses in higher years.

The cost of the accelerator mini-course was £1400 for paying facilitators plus £400 for the follow on sessions not originally designed for; the staff organiser had an additional 20 hours contact time costed at roughly £1,000 under FEC (Full Economic Costing), plus about the same amount again organising the scheme e.g. booking rooms, supplying the week's problems and other materials, managing the facilitators, etc.; plus a harder to quantify amount of work designing it in the first place. The overall cost that would recur each year if this became established practice (i.e. discounting the cost of designing the intervention and gathering evaluation data) would be about £4000. Thus

retaining a single student for a single additional year who would otherwise drop out would pay for the scheme nearly twice over from the institution's viewpoint.

Attendance data

There were 21 who formed the target "at-risk" group entering the second year computing course, having got less than a C grade in programming the year before, and six who opted to attend even though they had achieved a C grade in first year.

Attendance was recorded at every session in the scheme, for the 18 (after allowing for university holidays) scheduled slots for each student. There were 10 "non-attenders" who attended none or only the initial session; the 3 "low attenders" who attended less than half (5-8) of the subsequent sessions; and the 8 "high attenders" who attended more than half (10-17) the sessions. (Note then that all this effort and expense benefited only 8 students, costing about £500 per student.)

If attendance is voluntary, then it is a behavioural indicator of real student attitudes (as distinct from what they may say partly to please the listener). If, additionally, student opinion on what helps them is a good indicator (and it is the only indicator behind many decisions on modifying teaching delivery), then there is something valuable here at least for the 8 of 21 at-risk students who attended over half (10 or more) of the sessions (plus another 6 with good attendance who were not in the at-risk category).

From a practical, formative point of view, one of the big questions our evaluation failed to answer was why only half of the at-risk students attended. Because in this first year the intervention was designed only shortly before term began, it was not in the course documentation nor was the wider culture of tutors and past students aware of it: attendance depended entirely on the announcement at the enrolment meeting. We do not know whether more could be persuaded to attend, and how that might best be done; or whether on the other hand, those who did not attend would not have benefitted and correctly self-selected themselves out of the accelerator course.

The objective (exam marks) data

The most direct and narrow measure of any effect of the accelerator course (which was targeted at programming rather than computer science in general) was performance by

the "at-risk" subset (defined by those with lower than C grade results in the previous year's programming module) on the second year module most concerned with programming, and which ran in the first semester concurrently with the accelerator course. The exam results showed that of the "at-risk" target group as a whole in 2005-6 (21 students), 4 (19%) achieved a C grade or better in the year two programming course, and three of these attended at least 8 sessions of the accelerator course. This is better than any of the three previous years.

Table 1 Proportions achieving a C grade in programming (of those who completed the year)

Year	At risk	C grade	%
2002-3	37	3	8%
2003-4	28	4	14%
2004-5	18	2	11%
2005-6	21	4	19%
*2005-6 highA	8	2	25%
*2005-6 nonA	10	1	10%

As the table shows, non-attenders perform similarly to past years, when a few achieved improving to a C grade, but high attenders (although the absolute numbers are very small) show about twice the number doing so. Analysing only the high attenders is the best measure of the potential power of the intervention: it can't be expected to have had an effect on those who didn't attend. On the other hand, analysing the whole cohort of at-risk students addresses the practical management question of the effect of simply providing the intervention to the class the way we did. It is the difference between different stages of applied research: in testing a vaccine for instance, it is the difference between reporting on patients with a real need but for whom delivery was definitely achieved, versus the effect of a national programme with all the issues of how many it reaches in practice. If in subsequent years we manage to induce higher attendance, then we might see either the larger effect achieved more widely, or on the other hand practical factors might emerge that block the apparent promise of the intervention.

Even for the high attenders, it is far from a reliable remedy: of the eight at-risk who attended 10 or more sessions, six failed to get a C, although two did. The negative aspect is that there is still only a 25% chance of an at-risk student improving to the

stronger position for progression of achieving a C grade. The positive aspect is that this represents doubling the proportion from previous years of those who do this.

There are also other indications of benefit from the accelerator course. In general there are indications both of an association (Spearman rank correlation 0.39, $p < 0.45$ one-tailed) between attendance at the accelerator course and marks on the year two programming module, and of an improvement of the accelerator students relative to the other students. The 21 at-risk students were by definition the bottom 21 of 111 by rank in the class on entry (as defined by their marks the previous year in programming). By the end of the second year course, eight of the at-risk students (six of them non-attenders at the accelerator course) were still in the bottom 21 places; in contrast eight (all medium or high attenders) were above the bottom 38 ranking places in programming and were at worst within 6% marks of achieving a C grade; and the remaining five were in between. In other words, about a third were helped a lot, about a third improved somewhat relative to others in the class, and about a third did not improve and mostly did not attend.

The impact on the broader goal of retention is also visible in the high-attending group, although small numbers must make interpretation tentative. Only three (14%) of the 21 at-risk students achieved an average of C across all computing courses (as opposed to the programming module), compared to 57 (46%) in the class as a whole, which is no better overall than previous years' at-risk students: but these were all high-attending students, so the proportion of high-attending at-risk students achieving a C average is an impressive 37.5% which is much higher than previous years.

Table 2 Proportions achieving a C average (of those who completed the year)

Year	At risk	C average	%
2002-3	37	4	11%
2003-4	28	5	18%
2004-5	18	2	11%
2005-6	21	3	14%
*2005-6 highA	8	3	37%
*2005-6 nonA	10	0	0%

Finally, of those students who were not at-risk but opted to attend at least half the accelerator sessions, two failed to achieve a C or better, while four did.

The subjective (attitude) data

We interviewed an opportunistic sample of students at the end of the enrolment class at which the scheme was announced. An important student comment noted in the evaluator's report on interviews was "I thought universities didn't do this kind of thing" (expressing relief and approval that such a scheme was being put on).

We attended one of the last meetings of each group and conducted both a form of group interview, and a questionnaire with open-ended questions in. This approach meant we essentially only sampled students who thought the scheme worthwhile, and failed to get feedback from those whose behaviour suggested they had dismissed it early on as not worth attending.

The high-attending students, at least, clearly felt the scheme was engaging, valuable, and a major quality enhancement for them, as illustrated by these quotes:

Written comments:

- *"A drastic improvement in my study habits. I do a lot more work in the lab as well as at home."*
- *"I used to ask many fewer questions, and sometimes give up".*

From the evaluator's notes on a group discussion:

- *"Some said that the sessions allowed them to understand the lectures better; and that they were now able to ask more questions in lectures and tutorials."*
- *"One did say that others in [the] class now realised that these accelerator students understood pointers i.e. [understood this taught concept] better than the rest of the class."*

These comments give a little insight into the ways the process might benefit students. However since the sample was not representative, and the exam and attendance data

has a more direct bearing on the overall purposes of the intervention, we will not report further on the qualitative data.

Educational interpretation

The process of designing the intervention and then doing the evaluation has substantially modified our views on what the issues are.

The usual view, superficially supported by observation, is that programming depends primarily upon aptitude, not effort nor teaching. There is a great range of personal ability even among those already trained and holding permanent commercial jobs in programming (a factor of at least 100 in productivity between best and worst in a team is widely quoted), let alone among incoming students. On Masters conversion courses, it is not unknown for students with a first class degree in another discipline to fail the programming component and the whole MSc degree: general ability and hard work seem not to be enough even to scrape through. On the other hand, the teaching staff's own personal experience, like that of many successful computing students, is of learning new computer languages themselves without needing any personal teaching. There seem thus to be grounds for attributing student success or failure to the students' own aptitudes, and not to the teaching, just as you would not expect a deaf person to hear you if only you learned to speak better. On this aptitude view, the prediction would be that interventions such as the accelerator course could make little difference; that marks on a programming course largely indicate aptitude; and that students who attempt to continue with low marks are ill-advised. The exam marks results are not strong enough to disprove this, but do contain definite indications that other more tractable issues might also be important.

Another somewhat different view is that effort and a particular pattern of study work is a main determinant of success. Learning computer programming requires surprisingly long hours of practice: it is not something that can be done in an intense burst of work shortly before an exam. Neither is reading or listening to lectures a central aspect: it is doing examples or otherwise practising. Thus to an even greater extent than in many other subjects, it is time on task (Chickering & Gamson, 1987) that is likely to predict success. Hence too, student "engagement" in the sense of actually working on the subject, is crucial. This dependence on suitable study habits is consistent with Breen's work (Breen, 2002; Breen & Lindsay, 2002) suggesting that student success depended

on a match between a student's preferred study habits and the demands (rather than the conceptual content) of the subject they choose. Some student comments, and the association in our results between number of sessions attended and grade in the second year programming course, offer support for this.

Related to this, from the staff perspective, poor attendance, failure to hand in course work, not wishing to talk to anyone about the course are all signs of a problem, and likely to be predictive of failure. A part of our thinking at the design stage, consistent with this, was somewhat punitive: that the problem was to convince these students that they were at-risk and should be frightened of failing, and that this is why they should turn up to extra timetabled hours (five per week at first) and work harder. This attitude of ours was rather markedly disconfirmed in the interviews after the enrolment session by the considerable approval expressed by students for the provision of the accelerator course, including the rather plaintive remark quoted earlier, that it was perceived as support that was sorely needed.

Another strand of our thinking was based on our experience of PAL (Peer Assisted Learning) schemes. These have many potentially beneficial elements behind the basic recipe of regular meetings for client students on a course, run without a staff member by older students (<http://www.psy.gla.ac.uk/~steve/pal/>). One aspect is the supportive atmosphere of small groups organised to help each other where possible, and to draw on more experienced students when necessary. While the tone of each group varies a lot with the particular personalities of the facilitators, they clearly tend to promote both social and academic integration, which Tinto (1975) theorises as the essential factors in predicting student dropout or retention. The feedback we got from the evaluation made it clear that (at least for the high attendance students) the participants did find the groups had a strongly supportive atmosphere. While this gains student approval, and is consistent with aspects of the education literature, it is not clear from our results that this is a determining factor in these students' success or failure.

However questions about whether the participants felt they could have organised the meetings as study groups themselves, and so not really require facilitators, made it clear that another factor was very important, and indeed perhaps the most important. This was that, although much of the time in the groups was spent on individual work, and then if participants were stuck the next thing was to attempt to resolve it by peer discussion, nevertheless an essential aspect was having more experienced students

present to give expert technical input when necessary. Indeed, the students suggested that this was common: that the points that were sticking points for one were often so for the whole group, so that peer assistance was often technically (although not socially) ineffective. This draws attention to how in a subject where the important learning is largely done by solo work on examples, a student with a conceptual difficulty they cannot themselves resolve, can in fact not learn anything more until they gain access to expert help. While in comparison to some other courses, programming is heavily resourced with weekly meetings with tutors as well as the lectures, having to wait for a week before any more learning can be done would be a very serious barrier to progress regardless of motivation.

One view of this is that it is about improving student access to feedback, and the enormous importance of this to learning (Nicol & Macfarlane-Dick, 2006). On this view, these sessions gave students almost daily opportunities for feedback, and this could be expected to be much more productive than weekly feedback. Note however that this is not feedback in the sense of a learner needing judgements by teachers (e.g. marks and/or pointing out where something is wrong). In programming this is mostly self-evident, or generated by the computer. Instead, the crucial element seems to be explanations either of the key concepts or of what is wrong with the offending piece of code. Human (re)explanation, not judgement, is what is essential. In particular, it is likely to be explanation from another person: usually the original lecturer (or textbook) will have given their best explanation the first time. In cases where this is not effective, repeated access to the same source is seldom useful: what is wanted is the view from another mind. Senior students are not just cheap: they will have a different view, a different way of expressing ideas, and are closer to the learners' perspective.

What has been shown

The absolute numbers are very small, so a) this is not a huge smash-hit effect or it would be clearer even with small numbers; b) conclusions must be tentative. However multiple indications do all point the same way.

It looks as if there is a measurable beneficial effect for students who attended the accelerator course for at least half the sessions compared to non-attending at-risk students. This is indicated by the numbers of attenders compared to non-attenders attaining a C grade in the DSA2 programming module, in the substantial move up the

ranking in class even for those who got a D grade, and in the number getting a C average across the six computing modules. For the first two of the measures, there is also a measurable effect in comparison to previous years for the at-risk group as a whole (i.e. grouping attenders and non-attenders together).

Nevertheless we have to remember that the odds are still very much against an at-risk student rising to a C average and so having an unequivocal opportunity for honours computing: even after the full benefit of this intervention we can only estimate their chances at 25% which is half that of the class as a whole. It would seem that the aptitude factor remains a powerful influence, even if this intervention suggests that the teaching staff are not wholly powerless to help apparently weaker students improve their level of measured achievement.

Finally, there was one other clue in this trial. Deriving from the aspect of the original staff thinking that saw this as a "motivation" problem, the at-risk students were divided into two groups with different facilitators. In one of these groups ("the cosy group"), the emphasis was on a supportive atmosphere and avoidance of any note of confrontation. In the other ("the kicked group"), the students were generally not only set targets on work to complete before coming to the next scheduled meeting, but were each questioned every session on the degree to which they had met these targets. The at-risk students who achieved C grades were all in the "kicked" group. We are not in a position to draw a strong conclusion from this: these groups were not randomly but self-selected, and they met at contrasting times of day (9am and 5pm) which might well have an effect. However it does suggest that, while clearly greatly appreciated by students, the feeling of support by itself is not effective in raising learning outcomes, while a change to study habits and particularly to time on task, is. Taking the feedback and data as a whole, we are now less inclined to focus on the punitive metaphor than on the basic practice of a regular and inter-personal review of actual learning activity, and plan to institute this in all groups in the next implementation.

Acknowledgements

We would like to thank Mel McKendrick for donating her time to help with the evaluation.

References

Breen, R. (2002) *Motivation and Academic Disciplines in Student Learning* Unpublished Ph.D. Thesis, Oxford Brookes University.

Breen, R. & Lindsay, R. (2002) "Different disciplines require different motivations for student success" *Research in Higher Education* vol.43 no.6 pp.693-725

Chickering, A.W. & Gamson, Z.F. (1987) "Seven principles for good practice in undergraduate education" *American Association of Higher Education Bulletin* pp.3-7

Nicol, D.J. and Macfarlane-Dick (2006) "Formative assessment and self-regulated learning: a model and seven principles of good practice" *Studies in Higher Education* vol.31 no.2 pp.199-218

Tinto, V. (1975) "Dropout from Higher Education: A Theoretical Synthesis of Recent Research" *Review of Educational Research* vol.45, pp.89-125.