

**Death to Deadlines:  
A 21<sup>st</sup>-Century Look at the Use of Deadlines and Late Penalties in Programming  
Assignments**

*K.Becker, CPSC  
University of Calgary  
2500 University Drive NW, Calgary, Alberta T2N 1N4  
beckerk@ucalgary.ca  
403-220-5769, (fax) 403-284-4707*

**Abstract**

The practice of setting hard deadlines, sometimes with stiff penalties for late submissions has been the norm in many CS departments for as long as most can remember. This paper will examine the notion of submission deadlines as applied to programming assignments, in particular in the freshman year of a four-year computer science program. A brief outline of the common practices and the effect on novice students will be presented, followed by an alternative strategy that was implemented in a relatively small (i.e. 30 students) class in September 2005. Some initial student reactions are outlined along with known research from education which supports a more flexible approach such as this. The discussion that follows will outline the expected benefits from allowing greater flexibility in assignment submissions, the context necessary for successful implementation, as well as some of the drawbacks and difficulties. It begins by examining the place of deadlines in the curriculum and the effect this often has on students. Next, a highly successful approach from an unlikely source is reviewed to discover which elements might be appropriated and applied in a post-secondary learning environment. That unlikely source is modern video games. The approach used there is highly relevant to work that involves setting challenges, and to experiencing and accepting their associated consequences. Both of these conditions exist in a typical CS programming assignment.

**1.0 Introduction**

A quick search for “assignment deadlines” on several of the major research databases (ArticleFirst, Eric, Ebsco) turns up fewer than a dozen articles that mention assignment deadlines – and fewer than half of those feature the topic. A further search through Google Scholar uncovers more - slightly over 100 if the search is restricted by date to after 2000; and nearly 500 if the search is unrestricted. However, the vast majority of these ‘references’ simply mention assignment deadlines and do not discuss them beyond mentioning the need to inform students of them or to have students learn how to meet them. It would seem that the notion of flexible deadlines is one that is rarely considered. The author’s department is no different from the norm and the traditions for deadlines there have been virtually unchallenged since its very beginning – 30 years. The practice of setting hard deadlines, sometimes with stiff penalties for late submissions goes unquestioned. A common late policy in many institutions is to deduct a full letter

grade for each day an assignment is late or worse yet is the proclamation, “if it’s not in on time don’t bother handing it in at all”. When we were students, many of us were subjected to the same treatment, yet now as instructors, we appear to have forgotten about the effect this can have on students.

It seems we rarely consider why we set deadlines in the manner we do, we just do. In fact when faced with the news that all deadlines, besides the end of term, had been eliminated in the second introductory programming course (equivalent of CS102) in the fall of 2005, most of the other faculty in the department could hardly believe their ears. To confound matters further, students in this class were encouraged to resubmit their work for better marks as often as they wished – again with no penalty. Apostasy! Sedition, at the very least.

## **2.0 A Case for Considering Flexible Deadlines**

At a time when enrollment in computer science (CS) and information technology (IT) programs is low and in some cases still dropping, it would seem prudent to examine ways to increase the attractiveness of our programs and to increase retention of those already in program. Excessive workloads, competitiveness, and “dullness” are all issues cited by both males and females as making CS less interesting. [1] There are, of course many ways to address these issues, but one element that deserves attention is the setting of deadlines. Conspicuously, those who have considered flexibility when it comes to deadlines have generally done so in the context of distance studies, women, visible minorities, and other students considered to belong to “at-risk” categories [2-4]. Back in the mainstream, it is generally believed that allowing extensions on assignments mitigates against students’ learning how to budget their time [5]. While it is true that professional practice includes a need to learn how to budget time, as well as an ability to plan ahead and meet clear deadlines, it is less clear that merely enforcing hard deadlines in *every* class actually achieves the goal of students learning these things. This author’s experience has been that the majority of CS students leave work till the last minute, and that their behaviour in this regard when they become seniors is little changed from their behaviour as freshmen, which would imply that our methodology may not be as successful as we hope. It bears repeating that the argument is not that future CS professionals don’t need to learn how to budget time and gauge requirements, but that there is indeed no evidence to suggest that the persistent enforcement of hard deadlines actually helps.

## **3.0 Programming Assignment Objectives**

What roles do assignment deadlines play in the education of computer professionals? Perhaps a place to start is by identifying the primary learning objectives associated with novice programming assignments. One would assume that experience with the programming language being taught would be a primary objective. Gaining an appreciation for, if not yet a full understanding of various fundamental concepts and logical program structures being taught should also appear high on this list. Perhaps notions of sound design, clear documentation, appropriate error checking and error recovery might also be included. It is even likely that a great many other objectives would be listed before one finds any mention of adherence to deadlines. In fact, even a

detailed examination of the objectives listed in the course outlines of the 2001 ACM Computing Curricula [6] reveals no mention of deadlines. Why then do we seem so invested in this decades (centuries?) old tradition? As this author often tells her students, if an instructor has no answer to the questions, “Why are we doing this?”, and “What is it good for?”, then perhaps it is time to reconsider its inclusion in our curriculum. Aside from unsubstantiated claims that enforcing deadlines helps students learn responsibility, it may be that convenience, practicality, and simple tradition are all that keep us from considering alternatives.

How might we approach this issue if we began by considering time management and accepting responsibility as actual assignment objectives? We teach programming using various techniques: through graduated examples, by providing partial solutions, modifications to existing code, and by providing support through various forms of coaching. By its omission from deliberate instructional attention in CS courses and programs, we seem to imply that no guidance or support is required to help people learn to plan and budget time. Many business schools do recognize the need and teach time management as part of their curriculum. If we were to accept a similar position, and treat time management and deadlines as learning goals in and of themselves, we might approach them somewhat differently. Helping students learn time management techniques requires different strategies from simple control. Not imposing strict, often non-negotiable limits is one way to do this.

Evidence to suggest that students view assignment deadlines as an indication of when to work rather than an endpoint for work [7]. Hard deadlines are discordant with the view that students seem to have of their significance, and this implies that addressing this explicitly might have a positive effect. Flexible deadlines on the other hand, can be made to mesh with the students’ view of when to work. Gauging how long an assignment will take to complete will vary according to skill, style, and background. It also takes practice. Supporting students while they gain this practice is bound to be beneficial.

#### **4.0 Observations and Experiences**

Over a seven year period, the author taught the equivalent of CS101 six times, and CS102 twelve times. Both are 13 week courses, and both are required for all CS majors. The class size in the CS101 course varied from over 200 students to just under 60, and class size in CS102 varied from just over 150 to 29. Both courses typically have six assignments, with the first one or two being relatively simple, but with a steep increase in the level of challenge after that. In past years the value of the assignment component of the course has typically been around 20-30%, but in the last two iterations of the CS102 course this was increased to 50%.

Initially, very strict deadlines were enforced through electronic submission, and late assignments were only accepted in rare circumstances. The justification at the time was the large class size which made most other approaches impractical. The norm was that assignments were to be handed in on time, or not at all. Occasionally, class-wide extensions were given if it became clear that a particular deadline was unreasonable, but

this approach is rarely ideal as students who work hard to meet the original deadline almost always feel cheated. In later semesters, the policy was changed to allow for late submissions, but with a stiff penalty: i.e. one full letter grade per day late. In several other semesters, students were given “bonus points” for on-time submission. These bonus points were collected over the term for various things and used to move students up to the next letter grade if they were near a cut-off when the final grades were computed. Assignment due dates were set to fall on Fridays, and students were given the bonus points for submitting on Friday, but allowed to submit until the following Monday without being counted as late. Approximately 10-20% of students consistently handed assignments in on time and earned bonus points. A radically different approach was tried in the fall of 2005 as the author had a particularly small class of 29 students. This time there were no deadlines at all other than the “natural” one at the end of term. In addition, students were allowed to resubmit assignments as often as they wished and received feedback after each submission. This approach was modified in the subsequent iteration of the course because the enrollment was just over 100 students, and adequate teaching assistant support could not be guaranteed.

Several variations on submission requirements have also been employed over the years, and there appeared to be no difference in submission rates when students were required to submit at least four out of six assignments in order to pass, or when they are not explicitly required to submit any. Submission rates begin at 90% for the first assignment and decline to approx 60-70% for assignment six. In the class where they had no due dates other than the end of term the submission percentages were still almost the same as in classes where deadlines were imposed. Students reported that the greatest satisfaction with the deadline policies when they had set deadlines, but were given bonus points for on-time submission rather than penalties for being late. Students felt that the set deadline provided structure, even though it was not enforced, and the penalty-free “grace” period allowed those students who needed it the freedom to adjust their schedules somewhat when other courses got in the way. It also allowed students to take the time to “fix that last bug”. Although the rate of assignment submission did not appear to be affected, some degree of flexibility in deadlines, combined with the option to re-submit assignments resulted in a significant improvement in overall grades on assignments. The per assignment average increased from a ‘C’ to a ‘B’.

## **5.0 What Can Be Learned from Video Games**

“(T)he quality of student learning as well as the will to continue learning depends closely on an interaction between the kinds of social and academic goals students bring to the classroom, the motivating properties of these goals and prevailing classroom reward structures.” [8 p. 171] Relative assessment, and “curving grades” often leads to classroom cultures that can be seen as competitive ability games, which are primarily failure-oriented. In other words the motivational factor in this scenario is often one of failure avoidance than of mastery. Other students become obstacles to be overcome rather than potential allies. [8] In the twenty-first century, virtually every CS freshman student has had some experience with playing video games. Casual polls would indicate that digital games are what prompted an interest in CS for one third to one half of all first year

students. Many of these students have grown up with these games, and the challenge and reward structures typically found there are familiar and well accepted. Thus it should not be surprising that a model that approximates what one encounters in video games might result in better motivation and achievement than a more control-oriented tradition might. Such a model includes rewards for achievement; low, but increasing penalties for failure to meet set challenges, and multiple opportunities to try again. Goals are clear and criterion referenced so that one student's achievement is not tied to another's failure. Clear and often descriptive explanations of what needs to be done in order to meet requirements addresses the second part of the model, and flexible deadlines addresses much of the first.

## **6.0 Benefits and Consequences**

There are numerous other benefits to allowing some flexibility in how deadlines are set and enforced. Allowing submissions beyond the "official" deadline with minimal penalty clearly lowers some of the stress associated with the fear of failure. Students have reported that this tractability actually encourages them to stretch and try approaches they might otherwise not risk. The end result is that they learn more. Malleable deadlines help to provide students with a degree of control over their own learning, which in turn promotes ownership of and responsibility for the work they do. Many students actually work harder when they are left to take responsibility for their own work. When students have the freedom to arrange their workloads to suit their individual schedules, they also complain less and are less likely to plagiarize the work of others. Although no study data is available to substantiate this, casual interviews with students indicates that an oft-cited justification in cases of cheating is that students ran out of time to do their own. Finally, greater flexibility helps to promote both gender and cultural equity by recognizing alternative skills and attributes. [9]

Flexible deadlines do not come without cost, however. Freer deadlines result in submissions that are spread out throughout the term rather than at specific times, and this can complicate the grading process, leading to a higher workload for the marker. It must be recognized that marking 25 assignments all designed to solve the same problem all at one time will take less effort than marking the same assignments over the course of several weeks. Allowing re-submission of assignments increases the workload further, although not by much, as the majority of students will still only submit assignments once. Early indications are that only 10-20% of students actually submit an assignment more than once. Instructors, teaching assistants, and markers must be thoroughly familiar with all the assignments throughout the term because they are likely to have to address questions that get asked out of the usual sequence. Although this is less of an issue in more senior classes, additional attention must be paid to students by the instructor and teaching assistants to help them learn how to manage their time and cope with this unaccustomed freedom.

## **7.0 Conclusions and Recommendations**

After trying several variations on setting deadlines and allowing resubmissions, the conclusion is that setting some limits on both when and how often assignments can be submitted is both practical and beneficial for the students. The concept of offering "bonus

points” for on-time submission was well received, and the upper limit on the amount of influence these points could have kept them in perspective relative to the overall course objectives. Posted deadlines for assignments are necessary, and it was helpful to clearly indicate that the instructor and teaching assistants would proceed to the next assignment once a deadline had passed. Allowing a grace period where no penalties are allotted helped to increase the overall quality of the work submitted, but there is also a point of ‘diminishing returns’ when it becomes counter-productive to keep working on the same problem. Thus setting increasing penalties for late submissions sends a clear message that students should be proceeding to the next task. As long as the possibility of earning *some* marks exists, some students will persist on the same problem, so the retention of penalties prevents these students from getting “stuck” on one problem for too long.

The ability to resubmit assignments and a grace period where reward is given for on-time submission and lowered penalty results in higher achievement and a greater overall reported degree of satisfaction. When administered in this way, the emphasis appears to shift more towards achievement than failure-avoidance. This model allows people to build confidence, and frees students to act collaboratively. All of these elements in turn contribute to a sense of community within the class which has been shown to help increase retention in both male and female students.

## 8.0 References

1. Yasuhara, K. *Choosing Computer Science: Women at the Start of the Undergraduate Pipeline*. in *2005 American Society for Engineering Education Annual Conference & Exposition: The Changing Landscape of Engineering and Technology Education in a Global World*. 2005. Portland, OR.
2. Patton, M.A., *The Importance of Being Flexible with Assignment Deadlines*. *Higher Education in Europe*, 2000. **25**(3): p. 417-23.
3. Ashby, A.A.T.O.U.U.K., *Monitoring student retention in the Open University: definition, measurement, interpretation and action*. *Open Learning* 19, no, 2004. **1**: p. 65-77 (13 pages).
4. Jackson, M.D., *A Distance-Education Chemistry Course for Nonmajors*. *Journal of Science Education and Technology*, 1998. **7**(2): p. 163 - 170.
5. Walker, H.M., *What teachers should, can, and cannot do*. *Inroads: ACM SIGCSE Bulletin*, 2004. **36**(2): p. 20-21.
6. *Computing Curricula 2001: Final Report of the Joint ACM/IEEE-CS Task Force on Computer Science Education*, E. Roberts and G. Engel, Editors. 2001, IEEE Computer Press: Los Alamitos, CA.
7. Drew, S., *Perceptions of What Helps Learn and Develop in Education*. *Teaching in Higher Education* 6, no, 2001. **3**: p. 309-331.
8. Covington, M.V., *Goal Theory, Motivation, and School Achievement: An Integrative Review*. *Annual Review of Psychology*, 2000. **51**: p. 171-200.
9. Scantlebury, K., *A Feminist Pedagogy in Undergraduate Science: Conflicting Concepts?* in *Transforming undergraduate science teaching: social constructivist perspectives*, P. Taylor, P.J. Gilmer, and K.G. Tobin, Editors. 2002, Peter Lang: New York. p. 117-143.