

## Teamwork: Tips and Suggestions

**Distributed:** Monday, January 3rd.

---

### Motivation

Until this point in your academic career, you worked primarily independently, and on projects of very limited scope. Once you are employed as a programmer, you will rarely work independently on a project again. A skilled programmer can only turn out an average of 10 lines of well-designed, documented, and debugged code per day. With most systems programs and larger applications requiring many thousands to hundreds of thousands of lines of code, these are clearly beyond the scope of a single hot-shot programmer; the time to market would be too great. Hence working with a team to produce a major software system is an essential part of being a computer scientist. The stereotype of a computer programmer as a loner who communes with his/her machine to avoid people could not be further from the truth. Professional programmers spend more time in design meetings, in code walk-throughs, communicating with other programmers, with users, with the system maintainers, with marketers, etc. than in front of a monitor. That should be your experience in this course as well.

One problem with working as part of a team, and working on very large software systems, is that the program is too large for any one programmer to understand the whole system. A software system such as an operating system has more components than a Boeing 747, and it is clear to us that no one person understands each and every component of a 747, much less their interactions. Hence adherence to good software engineering practices is essential if the results of a large group programming effort are ever going to work together, be debugable, be maintainable, modifiable, or meet the original requirements.

Working with a team can be anywhere from fun to awful, depending on your attitude and the attitudes of your teammates. Since you may not know the work habits or attitudes of your new team members how can you insure a successful project and fairness in grading? Everyone needs to be involved in each aspect of the project (design, documentation, test planning, coding, and testing.)

### Foundations

**Respect for each other.** In order to work on a team you will need to be considerate of your teammates. They are all college juniors or above. They are capable, intelligent people and deserve respect. Treat your teammates as you expect (and deserve) to be treated, and hold them to that same standard.

**Fostering synergies.** One of the great strengths of a team is the opportunity to learn from one another and to develop and apply your respective skills to the problem at hand. You will probably find that different members of the team excel at different tasks. Match personnel with their skill set while remembering that everyone must get a turn leading the writing for a programmer's guide or user's guide and everyone must get a turn working on implementation.

## The 7 Habits of Highly Successful Teams

**Share contact information.** Exchange telephone numbers and email addresses. Make sure you remain available to your teammates! If you are leaving for the weekend, make sure your teammates know where you are, when you will be back, and (if possible) how to get a hold of you if they need to. Respond to their calls, messages, and emails! Even if the response is bad news, it is much better to be in touch than to be isolated.

Many team problems can be traced back to a lack of communication. Sometimes this lack of communication is due to simple logistical issues such as not checking one's email with the appropriate frequency.

**Meet early and often.** Group meetings are a valuable way to exchange information in real time. They can be much more efficient than extended email or IM discussions. Be careful not to spend all your time in meetings, but be sensitive also to the need for other members of the group to be current on your work (and vice versa).

**Plan.** Planning is the most important thing you can do. One hour spent in a preliminary team meeting saves many individual hours of redundant and possibly incompatible coding. Many students seem to think time planning and designing away from the keyboard is wasted effort. It will not be in this class.

You should all agree on a common language and a common hardware platform. Unless you are all extremely skilled (or masochistic), you should not use multiple platforms.

**Organize the group.** Many different organization for the team can be effective. Some teams work best when one person acts as a "manager", tracking progress and handling the partitioning of tasks. This person facilitates the group meetings to ensure they are focused and productive. Other teams prefer a flat structure, where members report to the group as a whole and the track each other's progress in group meetings.

Whatever structure you choose, make sure that everyone understands and accepts the structure. Also, you must rotate personnel around so that everyone gets a chance to be involved in all aspects of a lab by the end of the quarter.

**Evaluate peers and provide feedback.** Evaluate each other's performance during the quarter. A formal quantitative (and qualitative) peer evaluation is required at the end of every lab, but you should also carry out an informal evaluation at intermittent points.

**Identify problems early.** Try to anticipate issues that could cause the group problems down the road. An obvious example is when a part of the lab begins to slip behind schedule. Use a timetable with milestones that are fine-grained enough to allow the group to identify *early* when the schedule has begun to slip. Other examples of potential problems include inconvenient meeting times or personality issues that create tension.

Identifying potential problems early gives everyone a chance to do something about them.

**Confront problems early.** Just identifying a potential problem is not enough. You need to do something about it! You can approach your teammate directly or you can come talk to me. Usually the problem can be cleared up by a friendly, civilized, moderated, discussion.

But this has to happen early enough for people to adapt and for the team to benefit! I will not look kindly on a group that reports in the 9th week of quarter that problem X has existed all quarter long.

## Frequently Encountered Problems

Most team problems occur because a member of the team is over committed. This over commitment may be due to school class load, work issues, family issues, etc. It does not necessarily mean that they are lazy or stupid. However, if you feel that a team member is not attempting to contribute to the team, just let me know. We can have some friendly discussions and many times resolve the problem.

**I feel that my teammates are not doing their “fair” share.**

Contact the graders and/or the instructor. We will have a group meeting or individual meeting to determine exactly what the problem is. The earlier we discover and correct problems the more flexibility we have in making adjustments.

**My teammate is a coding “whiz-kid” and has decided to simply do it all by them self.**

Contact the graders and/or the instructor. We will have a group meeting or individual meeting to determine exactly what the problem is. The earlier we discover and correct problems the more flexibility we have in making adjustments. The “whiz-kid” who prevents others from working on the lab will have their lab grade reduced.

**Two of my teammates are long time buddies and they do everything together (software wise) and leave me out.**

Contact the graders and/or the instructor. We will have a group meeting or individual meeting to determine exactly what the problem is. The earlier we discover and correct problems the more flexibility we have in making an adjustment.

**What is differential grading? How can we avoid it and should we?**

If the graders and I determine that a fair share of the work was not done by all team members, then different grades will be assigned to each team member. If one team member does it all, he/she

may get a lower grade than the rest of the team. But no one will be happy with the grade. The team members and the instructor will have a meeting to discuss the problem and hopefully correct it. However a differential grade may still be assigned.

**Can I still pass the class without doing anything on the labs?**

Absolutely not.