Working with Norman Ramsey
A Guide for Research Students

Norman Ramsey
Tufts University

Spring 2016

Goals

We want to write the best software and publish the best papers in the best conferences and journals. We want our stuff to be the stuff people will read five or ten years from now to learn how to build their own systems. And whether we do compiler construction, language design, productivity tools, run-time systems, type theory, or some other project, we want always to focus on making the programmer's life better.

Introduction

Agreements between junior researchers (undergraduate students, graduate students, and postdoctoral associates) and their faculty supervisor (me) are often implicit. But an implicit agreement is little better than no agreement. This document makes things explicit. Much may be common knowledge, but writing down common expectations helps prevent misunderstandings.

If you have been invited to join my research group or are already a member, please read it all carefully. If you have questions or concerns, please talk about them; the document, like my research style, is a work in progress. I review everything periodically to make sure that my expectations and obligations are clear and that nothing important is forgotten.

In the spirit of full disclosure, I have tried to identify what's distinctive about working with me—both the good and the bad:

- Working with students and contributing to their professional development is the most rewarding part of my job. My ideal is to help *students* develop into *colleagues*. (I also enjoy working with other faculty.)
- I am ambitious and have high standards, and I expect the same from you.
- I can be disorganized, and you may have to compensate. You must always know what you're working on and what the plan is. If you don't know, ask.

- You can take on as much responsibility as you want. I hope that even students who start out doing "only" programming[1] will eventually become skilled researchers: people who can identify important problems, solve them, write about them, and present them to a technical audience. To that end, I seek opportunities for you to travel, publish, and present.
- I am obsessively precise.
- I use a sophisticated mix of software tools and programming languages, and I expect you to do so as well.
- As a senior professor, I have many other calls on my time—from classroom teaching to curricular planning to helping run the university. Compared with a more junior professor, I am more experienced and better connected, but I have less time for you and more distractions to contend with.

  That said, I want to work with you directly—that's why my research group will always be small.
- External commitments have made me skeptical about new technologies. If I'm lucky, I can afford to master about one new technology each year.

What's it all about?

Here are some of my perspectives on research and research students. Graduate students undertaking doctoral study are here to learn to become independent researchers—as soon as possible, you, not a manager or a supervisor, will decide what is the best use of your abilities. Undergraduate and master's students are more likely to be sampling research as one of a number of activities. In both cases, doing research under the supervision of a faculty member offers an opportunity to develop advanced skills while contributing to improvements in engineering and technology. Here are some of the elements:

- Although a doctoral student may begin with an "immigration project" in which he or she works for me, my goal is that each doctoral student should, as quickly as possible, be working for himself or herself, to advance his or her own professional goals.
- You must be aware of the research literature related to your project. Beginning students can expect substantial help from an advisor, but if you are undertaking a PhD, the expert on the literature will be you, not your advisor.
- To be successful, research must be documented. In the excitement of the moment it is easy to forget or omit vital details. *Contemporaneous* notetaking helps you remember, organize, and communicate your results. You must keep a notebook. (Electronic notebooks are OK, provided stuff is timestamped.)

[1] As if that were easy.

- Graduate research requires full-time dedication and year-round effort.[2]
- The field progresses only when good work is published. Publication is also a universally acknowledged sign of personal success. Writing for publication is an essential part of the graduate research experience; a typical dissertation will be connected to several submitted papers.
- Most published papers are presented at conferences. With my help, you will learn how to plan, prepare, and deliver an effective presentation.
- Undergraduate students are not required to write for publication, but if you want to try your hand, I offer every encouragement, including paid travel to conferences at which you can present a paper.
- The best researchers, although independent, don't hesitate to get advice from others, especially in areas in which they are not the world's experts. Even if you are just starting, take initiative and discuss ideas and problems with other students and faculty, not only within Computer Science or within Tufts, but also outside. In my field, programming languages, we are lucky to have world-class colleagues at several local colleges and universities.

Postdoctoral study

If you've completed your doctorate, the world assumes that you have the basic skills of a researcher: to identify and solve significant problems. Why do a postdoc?

- If your ultimate career goal is to become a professor, a postdoc offers you an opportunity to *spend time doing only research.* You can spend one to three years solving interesting problems without worrying about teaching, advising, or committee work. If you want, you can even be insulated from writing proposals.

  I encourage you to view your postdoctoral years not necessarily as a time to improve your publication record, but as a time to *develop intellectual capital* that you will draw on for the first several years of your next position.
- No matter what your ultimate career goal, a postdoc offers you an opportunity to *learn new ways of doing things.* You can use a postdoc to learn new experimental techniques, new tools, new ways of thinking, new ways of teaching, new ways of running a research group or a department, even a new country and culture. A good postdoc should enrich you, and you should leave with lots of new tools in your intellectual toolbox.

[2] Part-time PhD study is possible but can be ten times more difficult than full-time study, especially during the crucial final year of writing the dissertation. I am willing to work with part-time PhD students, but only those who demonstrate unusual maturity and ability to master new skills.

I have two primary expectations of postdocs:

- You must contribute to the research mission of the group. This means doing good work and publishing it. There should be a plausible relationship between your work and your source of funding.
- You must take responsibility for the success of your postdoc. Only you know what you want to take away, so only you can set goals and priorities. My job is to assist you every way I can.

Basic work rules

I expect good work habits, organization, and time management.

- Working at odd hours and in odd places is more common than a regular schedule. But to create opportunities to interact with colleagues, I expect everyone to spend *some* regular time in the lab—at least several hours each weekday. Many people in our group prefer to work by themselves in the morning (e.g., on writing) and to work interactively in the afternoons. Some remote work (telecommuting) is acceptable, but you must also spend time in the lab so you can interact with others. *Part of your job is to be in the lab and talk with other people about your research.*

  This short blurb may give you some insight why facetime in the lab is so important:

  > I wonder how an English professor would feel spending a week in a physics lab. Not about the scientific work, but about the frequent, ongoing interaction between students and peers, postdocs and faculty. Scientists see each other in the lab, if not daily, then at least weekly. They have frequent lab meetings, colloquia and interaction with scholars at other universities around joint research. During my graduate training in psychology at McGill University, especially in the research lab at the Montreal Neurological Institute, I spent hours hanging around the postdocs. I learned at least as much from them as I did from my interactions with my professors. The expectation was that I would be at the lab 9 to 5 or more, every day. I saw my adviser every day.
  >
  > — Gina Hiatt

  Ideally, my group has a short briefing every Monday through Thursday from 11:45 to 12:00. (I have a sleep disorder which sometimes makes a daily briefing impossible.)

- I discourage "death from overwork," but at certain times (when conference or grant deadlines approach, or when a project's goals are not being met), I expect extra time (evenings, weekends). After a crunch, compensatory time off is not only acceptable but recommended.

- If you are an undergraduate student and you are working for me part-time, please figure out how much time you can commit. We will review your commitment every semester. If you find yourself overcommitted, please squeeze out a few hours so your work can be handed off to someone else, not lost. Remember, it is always possible to leave a project with honor, but it is never honorable to disappear without a word.
- If you are a graduate research assistant, plan on doing some work on your projects between semesters and during long academic breaks.
- If you will need time off, please discuss it with me in advance.
- Make sure I have your current phone number and email address.
- If special circumstances require you to miss work, let me know.

The practice of research

The key principles are to know *what you have done*, to know *what you plan to do*, and to know *what is going on around you*.

Research meetings There are many kinds of research meetings.

- In an individual meeting, you meet with a supervisor or supervisors. Such a meeting is a good time to get some opinions about your new ideas, to get help with problems, and to communicate with your supervisor about progress.
- Project meetings may involve several people working on the same schedule. They usually involve planning and status reporting. This may be a good time to let other people on the same project know what you are doing.
- Group meetings may involve one or more research groups with researchers at all levels of seniority. They are a good way to develop broad ideas about what people are doing and where the field is going. They may also present opportunities to develop important skills, such as presenting work in front of groups. Finally, group meetings help build a sense of community.

To get the maximum value from our meeting time, I usually plan frequent, short group meetings supplemented by individual and project meetings as needed.

Here are some best practices for research meetings.

- Know when the next meeting is.
- For an individual meeting with your supervisor, it is a fine idea to bring a *written agenda* the meeting. Have ink on paper.
- *Ask questions* during meetings. Don't take something on faith just because the person who says it is more experienced.

- *Take notes.* If you need time for notes, ask people to slow down.

- After a project meeting or a supervisory meeting, use your notes to *write down your understanding of the decisions made at that meeting.* As well as any decisions made, be sure to cover your *plan of work* for the time until the next meeting. When you have finished, *email your summary to me* for confirmation.

  Post-meeting email is difficult for people to carry off. I keep pushing it for two reasons:

  - By writing things down, you clarify your own thoughts about your plans. You also get a chance to get help tying up loose ends or solving mysteries right away. This kind of help might otherwise have to wait for the next meeting.
  - You create a permanent record of what you've been working on. This permanent record helps me write you a great letter of recommendation. By drawing on specifics from your emails, I can write a far stronger letter than the typical "Mary is a really bright student and you should hire her."

- I enjoy meeting with students, and I work with my door open. Please drop in for a spontaneous meeting at any time, excepting only if my door is shut. If I am too busy to meet, I will chase you away.

Records

- Get a *lab notebook* to keep a permanent record of your work, about which more below.

- Maintain an *annotated* bibliography of references useful to your project. BibTEX can be helpful here. Or if you can get it to do anything useful for you, try Mendeley.

- If you borrow a book from me, *leave a trail*: sign it out.

The lab notebook This is how I expect you to use your notebook:

- Every time you work on your project, write in your notebook the date and time and what you did. (If you are being paid an hourly wage, you have to keep track of your hours anyway.) Even a single sentence provides a useful record.

- If you take experimental measurements, write them in your notebook.

- Your record should include things that failed as well as your successes. Write down what it was that didn't work; you may save someone else many hours.

Big things that don't work should definitely be written down, but even little things that don't work are often worth writing down. If it's a good idea for Don Knuth, it might be a good idea for you. (See "The Errors of TEX," *Software—Practice & Experience*, 19(7):607–685, July 1989.)

- Your notebook is a good place to keep records of test inputs and outputs, transcripts, screen dumps, etc. Print them out and tape them into your notebook. In some ways, a notebook is better than a demo, because there's a permanent record.
- You may also find it useful to use your notebook to sketch ideas, observations, measurements, proofs, code, solutions to problems, or whatever. Put these things directly into your notebook, not on scraps of paper to be transcribed later. Go wild; notebooks are cheap.
- Don't let the dog eat your notebook.

Professional interactions

- Be aware of what others are doing, both in the group and out. This knowledge and habit will serve you well throughout your career.
- Offer your experience to help others.
- Seek the experience of faculty and other students to help yourself.
- Be active at meetings and conferences. *Ask questions.* Do so respectfully, concisely, and often.
- Review manuscripts, both within the group and outside. (This activity is optional for undergraduate students.) I will guide you.

Good citizenship I expect everyone to work together to identify and solve problems.

- When we review code, I expect everyone in my group to participate, even if the code is not directly related to their project.
- If you see a need, find a solution. I will help you implement it.
- I may ask you to undertake service work for the group or for our department.

Remote work There are times to work remotely and times not to. We all benefit by having a group of people we can talk with about problems. If we are all sitting quietly in our rooms, we lose the benefits of those interactions. But at times it can be better to work alone, without interruption.

Some hints:

- You can do background reading anywhere.

- When you're trying to understand a problem and sketch the shape of the solution, discussions are invaluable—especially when the people around you have their fingers on the same topics.
- Once you've reduced something to programming, you can do that anywhere you have a computer—but programming is difficult, and you'll do a better job faster if you talk with people about what you're programming.
- My time is chopped up into small pieces. If I unexpectedly get twenty minutes free, it goes to a person who is in Halligan.

Group and departmental citizenship

Good citizenship is valuable not only for the research group but also for the department. Please don't hide in your office; get out and meet people at group and departmental events. Everyone should attend one group event per week, and everyone should consider attending departmental colloquia.

- *Lunch.* Faculty, especially Tufts faculty, tend to sit in their offices and work through lunch. Discourage this tendency! Invite faculty to have lunch with you; bring your own or go out.
- *Fun in the afternoon.* While classes are in session, people who are interested in functional programming meet once a week (the schedule may change semester by semester). We typically look at someone's code, write new code of our own, or discuss a paper. Occasionally someone gives a talk or a demo about some interesting technology. We usually meet for about an hour and a half. The *Fun* group has a wiki at `http://fun.cs.tufts.edu`.
- *CS colloquium (Thursday afternoon).* The Computer Science colloquium is held every Thursday from 2:50 to 4:00 in Halligan 111. Refreshments are served at 2:50 and the talk starts at 3:00. A good colloquium is a great opportunity to broaden your education by learning new things from people who aren't normally here. A colloquium that is not so good can at least teach us something about how not to give a talk. If a colloquium talk is too specialized for a general CS audience, complain to the host!

  For undergraduate students, there is probably not much benefit in going to every colloquium: the talks tend to assume a solid undergraduate knowledge of computer science. But do keep an eye on the speakers, and if you see a talk that looks interesting, come.

  For graduate students, it's not always obvious why you should go to colloquia. The benefits are primarily long-term and indirect.

  – You build up knowledge of a broad set of areas in computer science, including many areas in which you may never do research. While you're in graduate school, this kind of breadth won't matter much,

but when you start looking for your first job, it will be terribly important. Especially if you are looking for an academic position, you will have to talk to many members of hiring committees who are not in your area. If you can talk with them about their own areas and can get excited about some aspect of what is going on there, they will be much more likely to want you to come.

- When colloquia are full of people listening and asking good questions, it helps visitors learn that there is a vibrant CS community here. Building Tufts's reputation helps everyone and ultimately increases the value of your degree.
- It probably won't happen very often, but every so often you'll get a good idea from a completely unexpected direction.

If a colloquium is given in an area in which you have a research interest, sign up to meet with the speaker. Most speakers enjoy meeting with students, and faculty from other institutions are always thinking about recruiting. If some visiting senior person comes in and asks you about your work, don't be shy! Explain what you're doing and why, *use the whiteboard*, be clear, and you'll be remembered during interview season.

Software tools

My group uses software tools and languages aggressively.

- *Literate programming* is essential for any software that more than one person will work on, or that will outlive its author's tenure with the group. I naturally prefer the Noweb tool, but nuweb, CWEB, FunnelWeb, and other variants may be acceptable.

  Noweb code should be *at most 88 columns wide* so the output can be printed. Blank lines inside a code or documentation chunk can be Good, but blank lines around a code or documentation chunk are Bad.

  `http://www.cs.tufts.edu/~nr/noweb`

- All work should appear on a server that is backed up *nightly*. The Unison file synchronizer can be used to keep a server consistent with a laptop or a personal machine at home. Another option is to keep all your work in a Git repository on the file server, and to push daily.

  `http://www.cis.upenn.edu/~bcpierce/unison`

- Sources, documentation, test scripts, and everything else edited by human fingers should be kept under source control. In 2009 and 2010, we experimented extensively with CVS, Subversion, Darcs, Mercurial, and Git. We've settled on Git, and you will need to learn it. It is a terrible tool, but the others are worse. *Code you commit to the* `master` *branch of a shared repository should compile and pass regression tests.*

  `http://www-cs-students.stanford.edu/~blynn/gitmagic/`

Commit and push your work often—if possible, every day. (If your code is experimental or broken, *create and push a new branch*.) If you don't plan to commit and push, it's easy to let a whole semester go by without committing your work. If other people don't see the work and can't use it, there's less chance of it being carried on after you go.

Consider using the Github issue tracker—our department has its own instance of Github, and you can get as many private repositories as you like.

- In the religious wars over editors, I take no position. I was indoctrinated into emacs at a time when vi was not really a viable alternative. I use emacs, but I'm not happy.

  If you use emacs, I do recommend the Magit interface to git—it provides a much better visualization of the state of your repository, and it makes common tasks very easy.

- To build internal software, we use `mk`. For external distributions, we use `make`. Makefiles must be simple; GNU extensions are not acceptable. GNU autoconf is evil and wrong.

  `http://swtch.com/plan9port/unix/mk-with-libs.tgz`

- For building tools written in Standard ML, we have an uneasy truce with SML/NJ and its Compilation Manager, which we extend as needed. We prefer Moscow ML and MLton, but for large programs, they aren't always practical.

- We use LaTeX and BibTeX for documents.[3] God help us.

- I encourage you to use `nbibtex` (`http://tinyurl.com/685y7r`), which helps work around the sheer ornery arbitrariness of standard bibtex keys.

- Standard ML code should use the capitalization conventions of the SML '97 initial basis:

`all_lower`	for types and type constructors
`ALL_CAPS`	for signatures and datatype constructors
`mixedLower`	for functions and values
`MixedUpper`	for structures and functors

  Never use `open`.

- Objective Caml code should use the Caml capitalization conventions:

`all_lower`	for types, type constructors, functions, and values
`ALL_CAPS`	for signatures (normally just `S`)
`Capital_words`	for structures, functors, and datatype constructors

[3] The "Collection of Computer Science Bibliographies" is a useful source of BibTeX entries.
`http://liinwww.ira.uka.de/bibliography`

Objective Caml code avoids identifiers with `InternalCapitals`.

As in Standard ML, never use `open`.

- Haskell conventions are so far Out There that I have only this advice: your code must compile without warnings, and pattern-match warnings should be turned on. `http://stackoverflow.com/questions/1983047`

- C code should be strictly ANSI conforming, without `#ifdef`. Naming conventions should be those of Hanson's *C Interfaces and Implementations*. Any consistent and readable layout style is acceptable. Major builds should be tested with all available C compilers (e.g., `gcc`, vendor's `cc`, and `lcc`).

- Our scripting languages of choice are `ksh` and Lua. We also tolerate `sed` and `awk`. Bash, Icon, Javascript, Python, and Ruby are discouraged; Csh, Perl, and PHP are not acceptable. (I've experimented with all these alternatives, and I will be happy to discuss my thinking. It boils down to a combination of expressive power, ease of learning, and longevity.)
  `http://www.lua.org`

- As our development environment, we use Debian Linux. It enables us to use a huge range of tools without creating a systems-administration nightmare. You may be able to get by with a derivative such as Ubuntu. At minimum, you will want to learn package-management tools. I use `aptitude`, which has a confusing user interface, but it does a good job of updating your machine without breaking it. If you want, I can give you a short tutorial.

  If you are already a guru of your distribution of choice, that works too.

Authorship

In the academic world, authorship is the most important form of credit. All researchers employed in whatever capacity should be able to expect authorship credit for their contributions. Here, in no particular order, are some thoughts about authorship.

- If you are a student working on a project under my guidance, it's typical for us to write a joint paper of which you are the first author. I expect you to put in substantial work on the manuscript as well as the project.

  I have worked with students who identified and solved problems on their own, with little or no technical support from our research group or our shared infrastructure. When this kind of work is submitted for publication, the student is the sole author—even when I advise the student, it would be inappropriate for me to be a coauthor.

- For a major group project that spans several years, students or others may make significant contributions without taking on major responsibilities for the project. Most often these contributions take the form of implementation work. These contributions will be recognized with authorship credit, but such authors will typically be listed last.

- A student who joins a project already in progress might or might not wind up as a coauthor—it depends on the student's contribution.

- When two or more people share major responsibility for a project or a paper, it is usually obvious who should be the first-named author. If it is not obvious, it falls to the *other* authors to ask for clarity, e.g., by saying "I think you should be the lead author on this paper."

- If other things are equal, when a junior person works with senior people, the junior person should be the first author. When faculty work jointly with a student, the student should be first author unless there is a good reason otherwise.

- When submitting a paper, don't overlook past contributions of someone who has moved on; such people still deserve authorship credit. Forgetting such a person is a big mistake, and people will remember it for a long time.

- An accepted paper usually comes with an obligation to give a talk. If a paper has multiple authors, normally the first student author gives the talk—with my help and support. However, if you have graduated and moved on to a non-research career, or if you have family obligations that militate against traveling and giving talks, I am willing to give a talk about work we have published together.

Undergraduate research can be different

If you're doing summer research for pay, then you're a full-time researcher, like a doctoral student only with less experience. But if you're doing a senior honors thesis or an independent study, strange things happen: you have a hard deadline, and when the deadline arrives, you've earned a grade.

Because undergraduate research stops when school is out, we rarely have the luxury of judging a final outcome. So the *process* matters. I consider these questions:

- When we meet, have you made progress since the last meeting?

- When you have made no progress, do you cancel meetings so as not to waste our time?

- If you think you have solved a problem, can you write a description of your solution in clear English?

- If you write some implementation, is it fit to be read by another person?

Outcomes, even if not final, count heavily in your favor:

- Have you solved any interesting problem?
- Have you created a piece of software that either embodies a solution to an interesting problem, or that advances the community's infrastructure to a point at which someone else might use it to solve an interesting problem?
- Have you mastered an intellectually coherent body of knowledge with which you were previously unfamiliar?

By these criteria, you can rate very high without actually having solved any of the problems you started out with.

Past students I've worked with have fit one of two scenarios:

- The answers to all the process questions were positive, and there was at least one positive outcome. These students earned A's.
- The student disappeared for most of the term and reappeared a week before the deadline, bearing a document. When confronted with the F he had earned, the student actually did some work. The student earned a D.

I can imagine various intermediate scenarios, e.g., you solve an interesting problem by week 5 and then disappear. Or worse, you claim to solve a problem and nobody can understand the solution. But I don't worry about these unlikely scenarios.

The life of the doctoral student

If you're a graduate student, you'll spend some time adding depth and breadth to your knowledge of computer science, but the main thing you're here to do is become a researcher. A good researcher needs three critical skills:

- To solve a problem that has never been solved before
- To identify what new problems are interesting enough to be worth solving
- To describe a problem and its solution clearly and convincingly, both in speech and in writing

A big part of my job is to help you find opportunities to work on problems, to talk and write about them, and to watch other people doing the same.

Working on and identifying problems By the time you graduate, how many problems should you have worked on? New PhD's who are in demand on the job market have usually worked on at least three.

- When you first come in, we'll try to find you an "immigration project" where we see a clear problem and have a pretty good idea what the solution looks like. This might be a small problem we have a kicking around or a larger one on which work is already in progress. With luck, this problem will get you a quick taste of research and a publication.

- After an immigration project, different students take different paths. If you know what you want to do next, great. If not, we'll hunt for a problem that looks interesting, where we believe a solution exists, for which you can take primary responsibility.

- The final stage is for you to find your thesis problem. Finding a thesis problem is, in my opinion, the most difficult part of graduate school. Finding a good problem is always difficult, but for a thesis problem, there are special constraints. Although we don't have to know what will come out, we have to be confident that *something* will come out. You have to do something solid relatively quickly (e.g., two years), but your thesis work should also be in an area you can continue to work on afterward.

To help identify good problems, I rely on my peers. Whenever I travel to give a talk or go to a conference, I find lots of people to talk about ideas. By hearing others' ideas and talking about my own ideas, I develop my sense of what people think is interesting. You will learn to do the same—it is an important reason to go to conferences.

Speaking and writing When you look for your first job, and later if you try for academic tenure, you will be known by your conference talks and your published papers. When I prepare a talk or paper, I remember that there will be people out there who will know me *only* by that one talk or paper. (This approach is one reason I publish comparatively few papers. I try hard to publish them in the best places, because getting a paper ready for a small workshop is almost as much work as getting it ready for a top-flight conference.) Especially in the systems areas, Tufts has relatively few resources to help you prepare talks and papers.

- One resource is a course I sometimes teach called *The Engineering Method of Technical Writing*. You can find its handbook on my web page.

- If your work is related to functional programming, *Fun in the Afternoon* is a good place to get feedback on a talk or on a draft of a paper.

- In programming languages, Tufts has too few students and faculty to sustain a strong weekly research meeting with talks. This situation should improve in Fall 2014, when Kathleen Fisher arrives.

- When a student gives an external talk (conference, workshop, visit, or whatever), we start by working together on the presentation. The next step is a focused practice talk with detailed feedback. We also invite selected people from other groups at Tufts. These practice sessions can be grueling but invaluable.
- On my web page, I've collected some suggestions about preparing and giving talks.
- For questions of style and usage, I turn to the *Chicago Manual of Style* and to the fine books by Fowler (*Modern English Usage*, in the *second* edition, not the third) and Garner (*Modern American Usage*). Every writer has pet peeves; my primary peeves are that I insist on the correct use of the hyphen and on author-date citations. I also frown on the use of the word "this" as the subject of a sentence.

Sponsored research

Funding for research students (as well as my summer funding) normally comes from "sponsored research." This category typically includes grants from government agencies like the National Science Foundation, contracts with industry or with agencies like DARPA, and the occasional outright gift from industry. Different agencies want different things in return for their money. DARPA, for example, often wants software, documents, and other "deliverables" as well as command performances at conferences and other meetings. I try to get most of our funding from NSF, because what NSF wants is topnotch research results, pure and simple.

Part of your job as a research student is to help make sponsored research successful. Ronald T. Azuma has written at length on this topic, in his essay "So long, and thanks for the PhD!," and I quote him here.

> Academia is a business, and "graduate student" is a job title. This is especially true at private universities. Academia is very peculiar type of business. It is certainly not the Real World and does not work in the same way that the ordinary corporate world does. However, it is a business nonetheless and as a graduate student, you must treat it that way. Graduate school made a lot more sense and became much easier for me after I realized this. If you think of graduate school as an "Ivory Tower" free of politics, money problems, and real-world concerns, you are going to be severely disappointed. If you don't believe me, read *The Idea Factory* by Pepper White for one account of graduate life at MIT.
>
> A few graduate students are independently wealthy or have fellowship and scholarship money that cover all their expenses for their total stay in graduate school. Such students are rare, however. Most of us needed financial support, in the form of Teaching Assistantships

or Research Assistantships (RA's). In general, RA's are more desirable to students since those can directly fund the research you need to finish.

Where does the money come from to fund RA's? Your professors have to raise funds from external organizations. These include government agencies such as the National Science Foundation (NSF), Defense Advanced Research Projects Agency (DARPA), the Office of Naval Research (ONR), and others. Private companies also fund some university research, although this tends to be less common, in smaller amounts, and in the form of equipment donations. These organizations don't just give money away as charity. They expect their money to accomplish something. Increasingly these days, this takes the form of a contract for a working demonstration that must be shown at the end. That means once the money is delivered, your professors must come through with the working demonstration. It is rare that they do this by themselves. Instead, they find some very capable, young, self-motivated people who are willing to work long hours for small amounts of pay. In other words, they fund RA's.

The RA job is crucial to the academic business. If the RA's cannot successfully conduct the research, then the demonstration will not work in the end and the funding agencies may not be happy. They may choose not to fund your professor in the future, which will bring his or her research program to a halt. And there are many professors and other researchers chasing too few research dollars these days; it is a competitive market. Thus, each professor wants the best students available. These students are the most capable ones who can get the research done required to fulfill the funding contracts.

That means you must treat an RA like a job. You must prove to your professors that you are capable of getting the work done, being a team player, communicating your results, and most of the other characteristics needed to do well in regular jobs. That's why many of the upcoming sections in this guide sound like ones written for the regular workplace.

What do you get out of this? At the start, you may have to do tasks specifically related to the funding contracts. But eventually your professor must be flexible enough to fund your own specific research program that leads to the completion of your dissertation. Your stipend and tuition waiver should be enough to live on frugally without going into debt. You will learn the state of the art in your chosen speciality and conduct cutting-edge research on a subject that you find interesting and enjoyable. If you don't find this compensation sufficient, then you shouldn't be in graduate school in the first place.

The bottom line: realize that academia is a peculiar kind of business and the role you play in this enterprise. If you do your job well (and have good negotiation and interpersonal skills, as discussed in future sections), both your needs and your professors' needs will be met. But don't enter an RA position thinking that the computers, research equipment, staff members and other resources that you are provided with are your birthright. Don't take them for granted! Most of those exist only because your professors have been able to raise the money to provide those to you. In turn, you must fulfill your end of the deal by doing great research with those resources. If you don't do your job well, don't be surprised if your professors choose not to fund you in the future. They do not have to provide you with an RA job or let you use the computing equipment they acquired. And the student who has no funding, no tuition reimbursement and no access to required computing resources is the student who leaves the university that semester.

My responsibilities as advisor

Funding I cover wages, stipend, or salary, and also materials, travel, publication fees, etc.

Project definition I help with

- Goals and relevance
- Time span
- Approach
- Initial set of references for bibliography

Work environment I provide space, equipment, computer accounts, software, etc. I will also provide pizza to anybody who is working late in the lab and gets hungry. Just send me the bill.

Guidance I will

- Schedule frequent short meetings (20 minutes tops) with you and other students, typically three or four days per week
- Be available to schedule longer meetings at need
- Be available for unscheduled meetings and to help with software problems in the lab
- When new skills are needed, teach them, find other instruction, or develop a plan for self-teaching; and follow up to check that skills were learned
- Help plan experiments, coding, testing, and path of project

- Discuss results and analysis
- Help over roadblocks
- Constructively criticize writing and presentation skills as part of professional development
- Evaluate your work thoroughly and carefully
- Encourage when things go badly; praise when things go well
- Criticize only actions, not people
- Require challenging goals to be set and met
- Help frame questions, and give you opportunities to find answers
- Give you experience in interacting with sponsors
- Help you prepare for qualifying examinations
- At need, help you learn to work effectively as a teaching assistant

Publication I will

- Suggest when and where results should be published
- Provide opportunities to present results at national meetings
- Outline papers with you
- Correct or make suggestions about drafts (critical review)
- For undergraduates, write draft of paper (if desired)
- Help you respond to reviews of submitted papers

Proposal, thesis, or dissertation I will

- Provide time and schedule for writing
- Critique student outline
- Critically review drafts
- Help form committee

Preparation for permanent employment I will

- Give you a view of the financial and administrative sides of research
- Help with job-hunting
- Write letters of recommendation[4]
- Nominate you for appropriate awards and fellowships

[4]If you are a doctoral student, I will not write a letter for you until I have seen the first draft of your dissertation. This policy protects you from trying to finish a dissertation on an impossible deadline, and more important, it helps ensure that when you go onto the job market, you are properly prepared.

Completing a thesis on time

When you write a thesis to earn a degree, the University imposes deadlines. *The real deadlines are earlier.*

- If you are completing a senior thesis, your committee must have it ten days before the University deadline.
- If you are completing a doctoral dissertation, your committee must have it three weeks before your scheduled defense.
- If you are revising a doctoral dissertation (post-defense) to meet a degree deadline, you must present the revised version for approval at least one week before your final copy is printed for binding. If your dissertation requires substantial revision, your committee or I may require longer to review the revisions.
- The arrival of a thesis or dissertation to review must not come as a surprise—let people know well in advance when they should expect to receive your manuscript.

Proposals

Whether it is required or not, a thesis proposal is a fine idea. A thesis proposal can head off potential problems and avoid having you invest lots of time in avenues of exploration that have low probability of paying off—or that have already been explored by someone else. A review of the proposal should also keep you from biting off more than you can chew. When a proposal is approved, it typically binds your faculty supervisor and committee, *not* you. That is, if you do everything in your proposal, we promise to grant you your degree, but even if you don't do everything in your proposal, you still might have done everything necessary to earn your degree. For example, as you undertake your proposed research, you may well identify more interesting problems or solutions that result in an even better thesis than the one you proposed—or you may find that a problem you proposed is intractable, while a related problem yields interesting results. These are good outcomes; it's all part of doing research.

The arc of the doctorate

One of the less comfortable parts of being a graduate student is the expectations are often unclear.[5] The expectations are unclear for a reason: the strongest influence over your doctoral study is the interaction between you and your advisor—but no two students are alike, no two advisors are alike, and no two doctorates are alike. But there is common ground.

[5] Don't worry. After you graduate, expectations will be even less clear.

- As you begin your graduate studies, you're still taking courses, and you're not really a doctoral student yet.

- Like most advisors, I expect you to get involved in research as soon as possible, and not to be distracted by trivia like course requirements and qualifying examinations.[6]

- As a doctoral student, you will become a relatively senior member of a small learning community. Rank has its obligations as well as its privileges: I will expect you to mentor younger graduate students and undergraduate research students. You may even be asked to help in the classroom.

- Facetime with me will vary depending on circumstances. At the beginning of your graduate career, you will get regular meetings and close supervision. As you get more experience, you will get more independent. You can expect extra help around the time of your qualifying exam and thesis prospectus, but the general rule is that after your first project, you'll get less time with me. (I'll probably be spending that time with a more junior student.) Then finally, as you complete your dissertation, your work will become a high priority—if you plan to defend your thesis within six months, you have the first claim on my research time.

  Against this background, there can be big fluctuations. When you come up with something good, your facetime with me skyrockets. And when I am shackled to a large required course, your facetime with me may plummet.

- How much is enough? Part of the answer depends on your career goals. In the aughties, candidates for interviews in academic or industrial research positions had *at least* three publications in top conferences like POPL, PLDI, ICSE, and ICFP. If you build that kind of publication record, you will have plenty enough for a dissertation.

  If you're not aiming for a top position after graduation, I still expect you to produce enough externally visible work to help support the students who will come after you, just as your work will have been supported by students who came before you. A good rule of thumb is to produce two publications in top conferences. Workshops and journals are OK, but our aim is quality, not quantity, and if all of your work is published in second-tier venues, something is wrong. Our goal is not to publish as many papers as possible; it's to do great work.

[6] When you are wrestling with your dissertation, you will agree that the qualifying exam is trivial by comparison. Or I will be the one buying dinner.

It's fun to have fun, but you have to know how

Research is hard, and graduate study is hard. Most faculty do a pretty good job of eliminating gratuitous difficulty, but there is plenty of difficulty left over that is an essential part of the enterprise. It is intrinsically difficult to do something that's never been done before, to achieve results that will matter to other people, and to explain those results so that others can understand them. You will work hard to become better and better at something that matters. I will work hard to help you find that something and to help you become great at it. To sustain yourself through difficult times, find research that means a lot to you. Successful researchers find interesting problems and fall in love with them. In operational terms, some Friday nights should be spent having fun out in the world, but if you aren't also spending some Friday nights having fun with your research, you're not doing the right research.

Aside from having work that you love, the other thing that will sustain you through your PhD is the relationships you will build with people who are here. Students in your cohort and in nearby cohorts are the people best equipped to understand what you are experiencing and what you hope to accomplish. Use your early years wisely, not only to get to know the faculty, but to get to know your fellow students.