

# Ethics in Computing

Rupesh Nasre.

HS3050 Professional Ethics  
January 2018

# Agenda

- Ethics in industry job
- Plagiarism
- Open source versus closed source

# Professional Life

- Following rules, mannerisms and supporting the **eco-system** of the company.
- Doing your job, and doing it well.
- Academia: How much of it do we do as a student?
  - Part of academic life is professional. More so in non-Indian universities.
  - Have you seen students who worked in industry before?

... I have gone through your webpage and I am keenly interested to pursue my internship in Machine Learning...

# Professional Life: Tasks

- ◆ Gathering customer requirements
- ◆ Prioritization of tasks
- ◆ Designing solutions
- ◆ Planning the project
- ◆ Getting approval from managers
- ◆ Assigning tasks to subordinates
- ◆ Sharing the plan with the QA team
- ◆ Monitoring progress
- ◆ Code-complete, giving to QA
- ◆ Prioritization of bugs
- ◆ Documentation
- ◆ Product delivery
- ◆ Customer feedback

**Deal with WHAT you do.  
These fetch you *salary*.**

# Professional Life: Responsibilities

- Dealing with **customer**
  - Are you listening?
- Dealing with **managers**
  - Are you backing up your prioritization with numbers?
  - Are you protecting your subordinates?
  - Do you understand why your manager is pushing for a feature?
- Dealing with **subordinates**
  - Do you respect the individual?
  - Do you have a career plan for each of them?
  - Do you lead by example?

Deal with HOW you do it.  
These fetch you *character*.

# What the customer really needed

How Projects Really Work (version 1.5)

Create your own cartoon at [www.projectcartoon.com](http://www.projectcartoon.com)



How the customer explained it



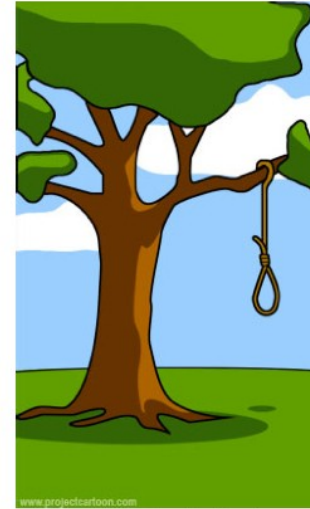
How the project leader understood it



How the analyst designed it



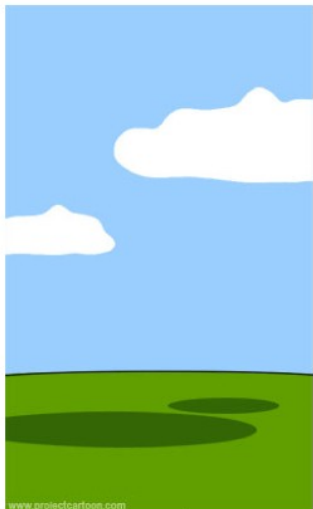
How the programmer wrote it



What the beta testers received



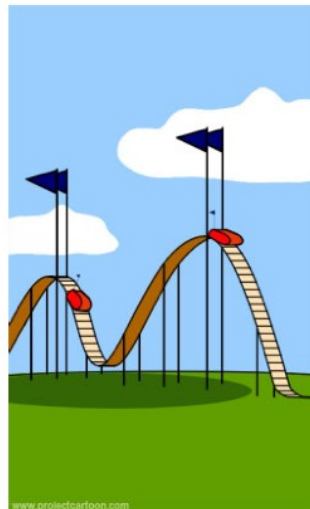
How the business consultant described it



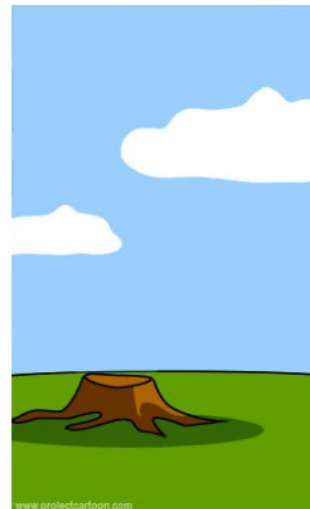
How the project was documented



What operations installed



How the customer was billed



How it was supported



What marketing advertised



What the customer really needed

# Responsibilities in Computing

- **Confidentiality**

- ideas, code, documentation, trends, weaknesses
- more so when you change the job to join a rival
- more and more so when you launch a startup

- **Security**

- Your code would run in client's production environment.
- The code may be used in mission critical applications.

- **SLA**

- The code should be bug-free.
- The code must meet the service-level agreement.
- **Anecdote:** German box-making company.

# Code Bugs

- Typically, developers and testers are not good friends.
  - It is a zero-sum game.
  - Developer wants to reduce the severity, and tester wants to increase it.
- **Acknowledge** the issue.
- If the bug is critical or difficult to find, **appreciate** the tester's hard-work.
- If you are overall happy with the code, appreciate the developer's hard-work.



# Other Professional Ethics

- *Thank you* -- especially, when you are dealing with a subordinate
- Being **punctual**
  - How many of you reach your classes on time?
  - How many faculty members end the class on time?
- **Replying** to emails
  - Within a deadline – typically, 24 hours
  - **Anecdote**: Experience with CE colleagues

# Question for Discussion

- You are appearing for placements. You are on medication for acute hypertension, which forbids you from working for hours or sometimes days when the attack happens.  
**Should** you mention this during the interview?

# Agenda

- Ethics in industry job
- Plagiarism
- Open source versus closed source

# Plagiarism

- Copying without permission / acknowledgment
- Why it differs from other disciplines?
  - Copying is easy!
- **Allowed:**
  - Codes on certain websites made explicitly available
  - Images permitted to be copied, but not modified
- **Not allowed:**
  - Inside an access restriction
  - Under a license
  - Nothing specified

# Repercussions of Plagiarism

- Academic setting:
  - DISCO, grade loss, loss of trust
  - moss, **anecdote**: crowdsourcing

# 35 said Copy, 2 said No

```
thrust::sequence(keys.begin(),keys.end());

for(i=0;i<N;i++){
    for(j=0;j<N;j++){
        std::cin>>a;
        inp[i*N+j]=a;
    }
}

thrust::sort_by_key(inp.begin(),inp.end(),keys.begin());
thrust::transform(keys.data()+N*N-N, keys.data()+N*N, div.begin(),
cols.begin(), thrust::modulus<int>());
thrust::transform(keys.data()+N*N-N, keys.data()+N*N, div.begin(),
rows.begin(), thrust::divides<int>());
thrust::transform(cols.begin(),
cols.end(),rows.begin(),ondiag.begin(),thrust::plus<int>());
thrust::transform(cols.begin(),
cols.end(),rows.begin(),offdiag.begin(),thrust::minus<int>());
thrust::copy(keys.data()+N*N-N,keys.data()
+N*N,cols_keys.begin());
thrust::copy(keys.data()+N*N-N,keys.data()
+N*N,rows_keys.begin());
thrust::copy(keys.data()+N*N-N,keys.data()+N*N,on_keys.begin());
thrust::copy(keys.data()+N*N-N,keys.data()+N*N,off_keys.begin());
thrust::sort_by_key(cols.begin(),cols.end(),cols_keys.begin());
thrust::sort_by_key(rows.begin(),rows.end(),rows_keys.begin());
thrust::sort_by_key(ondiag.begin(),ondiag.end(),on_keys.begin());
thrust::sort_by_key(offdiag.begin(),offdiag.end(),off_keys.begin());

for(int k=0;k<(N-1);k++){
    if(cols[k] == cols[k+1]){
        out[0] = cols_keys[k];
        out[1] = cols_keys[k+1];
        break;
    }
    else if(rows[k] == rows[k+1]){
```

```
thrust::device_vector<int>div(n,n);
thrust::device_vector<int>row(n);
thrust::device_vector<int>col(n);
thrust::device_vector<int>d1(n);
thrust::device_vector<int>d2(n);
thrust::device_vector<int>val1(n);
thrust::device_vector<int>val2(n);
thrust::device_vector<int>val3(n);
thrust::device_vector<int>val4(n);
thrust::device_vector<int>value(n*n);
thrust::sequence(value.begin(),value.end());
```

```
thrust::sort_by_key(board.begin(),board.end(),value.begin());
thrust::transform(value.data()+n*n-n,value.data()
+n*n,div.begin(),row.begin(),thrust::divides<int>());
thrust::transform(value.data()+n*n-n,value.data()
+n*n,div.begin(),col.begin(),thrust::modulus<int>());
thrust::transform(row.begin(),row.end(),col.begin(),d1.begin(),thrust::plus<int>
());
thrust::transform(row.begin(),row.end(),col.begin(),d2.begin(),thrust::minus<in
t>());
thrust::copy(value.data()+n*n-n,value.data()+n*n,val1.begin());
thrust::copy(value.data()+n*n-n,value.data()+n*n,val2.begin());
thrust::copy(value.data()+n*n-n,value.data()+n*n,val3.begin());
thrust::copy(value.data()+n*n-n,value.data()+n*n,val4.begin());
thrust::sort_by_key(row.begin(),row.end(),val1.begin());
thrust::sort_by_key(col.begin(),col.end(),val2.begin());
thrust::sort_by_key(d1.begin(),d1.end(),val3.begin());
thrust::sort_by_key(d2.begin(),d2.end(),val4.begin());
for(i=0;i<n-1;i++){
    if(row[i]==row[i+1])
    {
        con1=val1[i];
        con2=val1[i+1];
    }
    if(col[i]==col[i+1])
```

# Repercussions of Plagiarism

- Academic setting:
  - DISCO, grade loss, loss of trust
  - moss, **anecdote**: crowdsourcing
- Professional setting:
  - Legal action (e.g., apple vs. Samsung)
  - Loss of millions of dollars (e.g., 2.7 bn USD above)
  - Bankruptcy (e.g., Firestar Diamond Inc.)
  - Note that the personal loss is very small compared to the company loss. Hence **ethics** matter.
- Research setting:
  - Banning submissions
  - Credibility loss

# Best Practices

- Protect your password.
- Instead of giving your code, give your **help**.
  - **Anecdote**: stronger student helping the weaker.
- Discuss **ideas**, do not show the code.
  - Help in debugging.
  - Do not help your friend *get* the degree. Help him / her *earn* it.
- Ask yourself whether you are cheating the instructor, the system, or yourself.
- Learn to say **No** – with an apology.



# Question for Discussion

- This is your last semester, last course, last assignment. If you do not share your code with your friend, he is likely to fail the course and would not get degree with you and the class.

What **should** you do?

# Agenda

- Ethics in industry job
- Plagiarism
- Open source versus closed source

# Quiz: Whose email?

The feedback we have gotten from the hundreds of people who say they are using BASIC has all been positive. Two surprising things are apparent, however, 1) Most of these “users” never bought BASIC (less than 10% of all Altair owners have bought BASIC), and 2) The amount of royalties we have received from sales to hobbyists makes the time spent on Altair BASIC worth less than \$2 an hour.

Why is this? As the majority of hobbyists must be aware, most of you steal your software.

Most directly, the thing you do is theft.

**Bill Gates, 1976, OPEN LETTER TO HOBBYISTS**

# Open vs. Closed Source

- Traditionally, software has been closed source.
  - allows **businesses**, competition
  - most earlier software companies, server pages, ...
  - sometimes needed for **security**
- Since two decades, more proponents of open source.
  - FSF, Linux, ...
  - Reliant on the philosophy of *knowledge should be free*.

# Philosophical Debates

- Closed source means you want to hide bugs!
- Closed source means you want to monopolize!
- Closed source means you don't trust me!
- Advantages of open source
  - Someone else can fix the bug for you.
  - Others can contribute to further development.
  - Overall understanding of the software may improve.
- Advantages of closed source
  - I can protect my company.
  - I don't want anyone to tamper with the code.
  - The code is relatively more secure.

# Open Source in Research

- More conferences / journals asking for code.
- **Artifact** evaluation
  - Optional, encouraged
  - Earlier, only for accepted papers
  - Now, for all papers, but only positives
  - Future, paper and code are evaluated together
- Other advantages
  - Advances state of the art
  - Improves **visibility**, citations
  - Improves collaboration in scientific community

# Publications versus Patents

- Ideas are published to **announce** results and allow others to build upon them.
  - Primary goal is NOT to stamp the idea with your name.
- Ideas are patented to **protect** the creation from getting duplicated without consent.
  - Primary goal is NOT to improve state-of-the-art.
- Same idea can be patented as well as published.
- Value of a startup is in its new technology.
  - The technology can be protected with patents.
  - Until few years ago, software-based ideas could not be patented.
- **Anecdote**: infringement lawsuits.

# Other Ethics in Research

- Citations
- Acknowledgments
- Author order



# Question for Discussion

- You are the team-lead. Appraisal time is around the corner; you are likely to get promoted. A high severity bug is found in your team's code. You did not write that code.

**Should** you safeguard your position (by conveying that the fault happened due to the developer and the tester)?

# Same Three Questions

- You are the interviewer. Should the student convey his / her medical situation?
- After 10 years, how does your decision to help or not help your friend matter?
- Your father / son is the team-lead. What do you want him to do?