

**“i want 2 do
project. tell
me wat 2
do”.**

Shakthi Kannan

GNU Free Documentation License
September 2008

Version 1.2

WARNING

The views expressed here are those of the author (otherwise, why would it be here?). You are welcome to agree to disagree, well, whatever. Any copying or public performance of this presentation is permissible (yay!) thanks to the GNU Free Documentation License.

Before you
begin

...

~~Sir~~

~~Madam~~

Address people by their

First Name!

**No room for any
sentiments/
emotional
feelings.**

Be thick-skinned.

Use /dev/null.

Earn
your Living

Hunger for

Knowledge

so-called “student”

+

degree

! =

engineer

You **cannot** buy
knowledge.

Learn to

READ

READ

READ

Read

Mailing List

Guidelines!

Collaborative Development Project Websites

<http://sarovar.org>

<http://savannah.gnu.org>

<http://sf.net>

<http://freshmeat.net>

When you write to your mentor(s), mention:

- **Skills**

programming languages, tools used.

- **Domain of interest**

application/system/databases/web/embedded
et.al.

- **Project time-frame**

Also write something **about yourself**, any previous Free Software experiences, interests etc.

Don't make **ORDERS**

like these ...

“ i like to be a project member in your team. please send me the details”.

“ i want 2 do project. tell me wat 2 do”.

no interest

+

final year project

!=

Free Software
project

Have detailed
**discussion with
the mentor(s)**
before starting the
project work!

Communication
during
the project ...

Read

Free Software

**Communication
Tools**

English is the accepted language in the business world, and in the Industry.

If your English is terrible, get help.

All your Peter jokes in school/college are useless.

Welcome to the real world!

Get serious. Get help with your English.

Language

Reading Comprehension

Writing

- Bad grammar is acceptable.
- Atleast do a spell-check.

Spoken

Every weekend

have a

one-to-one

chat

discussion with

your mentor(s).

Mentor(s), who work, are busy during weekdays.

Don't disturb them during weekdays, unless it is urgent.

Just send
e-mail

during weekdays.

If your English is terrible, don't explain the error!

By trying to explain it in your own words, you only make it worse!

Just copy and paste the **output/error** or **log** messages.

Not being **on time** is

unprofessional,
rude,
awful!

Punctuality is important.

Time is precious!

As on date, you cannot undo time!

If you are not able to make it
to the online discussion with
your mentor(s),

inform!

Mentor(s) can get to do their
work rather than wait for you!

When asking a question
in writing, put a
question mark
at the end.

How does anyone know if you
are making a statement or
asking a question **?**

Doing the project

...

Understand what mentor(s) can do ...

Mentor(s) **CANNOT**

- write project abstracts,
- write code,
- do project documentation,
- prepare presentations,
... for you.

Mentor(s) can only tell you what to do.

Start with **small** tasks to
assess your progress,
before you handle **big**
project tasks.

Patience is the key to
success.

Send status
updates on daily
basis!

Never ever do
last minute work!

curve !
learning
a steep
through
going
You are

Communicate like an engineer!

Always give detailed info when posting a problem or asking a question.

Don't give **blunt** information like:

“Internet is not working”.

“I have compilation problem”.

Communicate like an engineer!

“Internet is not working”.

- What is the network setup?
- What did you try?
- How do you say it is not working?
- Which GNU/Linux distribution?

Communicate like an engineer!

“I have compilation problem”.

- What source code?
- How did you compile?
- Which compiler?
- Where is the error output?
- Which GNU/Linux distribution?

HOWTO report bugs effectively

<http://www.chiark.greenend.org.uk/~sgtatham/bugs.html>

Bug Writing Guidelines

<http://marianne.in2p3.fr/datagrid/bugzilla/bugwritinghelp.htm>

Keep journal/log of:

- **daily activities,**
- **e-mail correspondences,**
- **problems solved,**
- **HOWTOs/reference links.**

Helps in project documentation.

When you reply to e-mail or questions, reply to all the questions asked.

Don't skip anything!

Same when following HOWTOs, documentation, tutorials et. al.

Over-enthusiasm can cause you to make your own decisions.

Never make your own decisions!

Else, why do you need mentor(s)?

Always inform your team members, or consult your mentor(s) before doing anything new.

When sending e-mails,
don't forget to CC
all your team members.

If you
intentionally/unintentionally
forget to CC, your team will
lose trust in you!

If you keep
repeating
the same mistakes
again, and again, and again,
expect the mentor(s) to find
other developers.

It is normal for people to
move in, and out of project
teams.

Can my
friend(s) also
join
the project?

Only if they get selected by the mentor(s).

**Never ask personal
questions**

to your mentor(s).

It is none of your business.

Respect other peoples' privacy.

Write small code first, and send it to your mentor(s) for feedback.

Easy to **correct small mistakes** on daily basis, than to submit a chunk at the end of the week, and re-do everything. PITA.

**Never make
assumptions.**

Always give proof when
you make statements.

Never hesitate to ask questions.

If in doubt, ask.

No harm in making sure you have understood.

Being silent gives the impression that you have understood.

Feel free to have open discussions with your mentor(s).

The more you hesitate, lesser the communication, more the communication gap, and project at stake!

Impossible
is a word only to
be found in
the dictionary of fools.

~ Napoleon