

# Do-s and Don't-s in your PhD

Alexander Gelbukh

[www.Gelbukh.com](http://www.Gelbukh.com)

# Agenda

- Paper
- Explaining
- Don't build systems
- Research topic
- Reading

# Paper

Clear, complete,  
published in a good journal

A big problem of many students:  
good research, bad reporting

~~Whatever looks like a paper, is a paper~~

# “Paper is bad, but the work is good”

- **Biggest problem of some students**
- The only goal of research is reporting
  - to advice knowledge of humankind, not yours
- Great travelers vs. tourists
  - Marco Polo went, saw, and told the world
  - Tourist goes, sees, and enjoys it for himself
- You travel into fascinating world of science
  - to see for yourself (tourism) or to tell others?

# Don't

# Do

- Are you a traveler or a scientific tourist?



# “Paper is bad, but the work is good”

- If you explain good idea in a bad paper, you just give your idea to somebody else
- Somebody will re-write it and publish in a good journal
  - Columbus discovered America
  - Amerigo Vespucci was the first to tell public about it in an understandable form
  - After who *your* great discovery will be named?

# Form is often more important than contents

- What you publish is the TEXT of your paper, not the WORK behind it
  - bad paper about good work is a bad paper
- Train yourself to SEE writing and formatting problems
- Your paper, as your suite and tie!



# Don't

# Do

- Who is more skilled? But who will get the job?



# Form is often more important than contents

- How many people in the world can understand how clever your paper is
- How many, that it is poorly formatted?
- If the author can't well align the margins, center the figures and formulas, use consistent indentation...
- ... is he experienced enough to write something reasonable?

# Form is often more important than contents

- Margaret Thatcher's grandma:  
If something is worth doing,  
it is worth doing well
- How clever your ideas are doesn't depend on you,
- but how perfect your writing and formatting are, does!
- Never send papers not perfectly written and formatted

# Tough competition

- Journals compete
- Impact factor: # cites / # **papers**
  - only last 2 years, VERY tough
- Cites are benefit, papers are cost
- VERY tough competition
- World championship
  - just running does not make you Olympic champion

# Hand-made gift for mammy: default action is acceptance



# Coursework: formality.

## Default action is acceptance



Paper: competition.  
Default action is rejection.  
**Masterpiece**, or will not sell.



# Clean paper

- Does your paper look and read *clean*?
- In some cultures, cleanness is not a priority
  - not bad in itself, they have other priorities
- Western culture is very sensitive to cleanness
- The reviewers are from that culture
- Even if this is not in your culture's priorities, you have to train yourself to see the difference
  - as I train myself to use only right hand in India





Kandel Photography.com







The reviewers do perceive the difference.



Kandel Photography.com

# One idea per paper

- Paper is a theorem
- Title is the formulation, the rest is proof
- Whatever you can't put in the title, will not be noticed
  - worse, somebody will notice it and publish
- Two ideas, two papers
- If you can't formulate the message of the paper in its title, re-think it

# Keep reader understanding

- Lack of understanding breaks communication
- Lost! If I don't understand something, how can I be sure that I understand the rest?
- At any point, don't leave unanswered questions
  - guess what questions the reader will have
  - even if you YOU know that this is no issue
  - even if *later* it will be explained, but *here* he is lost

# Keep reader understanding

- Topological order of ideas / definitions
- Cannot answer it here? Make it explicit
  - we use lexical functions **WHAT'S THIS?!**
  - we use lexical functions (see page 23)
  - we use so-called lexical functions
- OK, I don't know what lexical functions are, but this is foreseen by the author. Can keep reading.

# Misunderstanding, worse than lack of understanding

- Foresee what the reader can MISunderstand
- If he does not understand, he knows this
- If he MISunderstands, he THINKS he does
  - keeps reading, re-enforces the misconception
  - misconceptions propagate
- Actively prevent misunderstanding
  - *Our approach is not a modification of SVM*

'Mine is a long and a sad tale!' said the Mouse, turning to Alice, and sighing.

'It is a long **tail**, certainly,' said Alice, looking down with wonder at the Mouse's tail' 'but why do you call it sad?'

And she kept on puzzling about it while the Mouse was speaking, so that her idea of the tale was something like this: —



'Fury said to a  
mouse, That he  
met in the  
house,  
"Let us  
both go to  
law: I will  
prosecute  
YOU. — Come,  
I'll take no  
denial; We  
must have a  
trial: For  
really this  
morning I've  
nothing  
to do."  
Said the  
mouse to the  
cur, "Such  
a trial,  
dear Sir,  
With  
no jury  
or judge,  
would be  
wasting  
our  
breath."  
"I'll be  
judge, I'll  
be jury,"  
Said  
cunning  
old Fury:  
"I'll  
try the  
whole  
cause,  
and  
condemn  
you  
to  
death."

'You are not attending!' said the Mouse severely.

'I beg your pardon,' said Alice: 'you had got to the fifth bend, I think?'

'I had **not**!' cried the Mouse very angrily.

'A **knot**!' said Alice, 'Oh, let me help to undo it!'

'I shall do nothing of the sort,' said the Mouse, walking away. 'You insult me by talking such nonsense!'

# The devil is in the detail

- Give **sufficient details** for complete, clear understanding
- Can the reader reproduce your experiments?
- (Can YOU reproduce them in a month?)

# Keep it simple

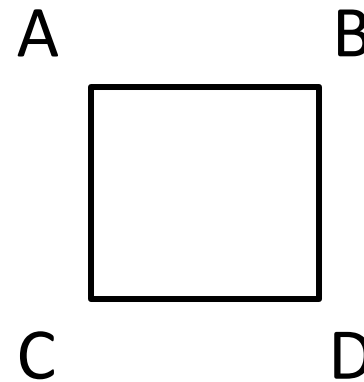
- *“Keep it simple, as simple as possible – but not simpler.”* Einstein
- A scientist takes what seemed complex and makes it simple
  - Ex: celestial mechanics / Kepler laws
- Why is science complex? Its object is complex
- Complexity is the enemy, not the goal!
  - ... *but not simpler!*

# Don't

- We consider a simplicial complex that consists of four one-dimensional geometric shapes of equal measure whose zero-dimensional cells are pairwise identified, along with the corresponding immersion in the standard  $\mathbf{R}^2$

# Do

Consider a square ABCD.



*This is better science.*

# Anchor in existing knowledge

- Written once, read many:  
it's *your* job is to make it easy to understand
- Anchor new things in known things
  - progress must be incremental
- Use existing terminology
  - devote effort to find it

# Don't

- I wrote a program
- It reads the data
- It puts the data points into space
- It uses examples to build a hyper-plane It assigns labels to the data, + and -

# Do

- I used supervised learning
- I used SVM classifier

# Explain differences

- If not directly used existing technique
- Explain the difference:
  - what you took
  - what you changed



# Don't

- I suggest a method
- It reads the data
- It puts the data points into space
- Examples, to build two hyper-planes
- It assigns labels to the data, + and -

# Do

- I suggest a modification to SVM
- Instead of one hyper-plane, it builds two

# Explicit contributions

- Reviewer, in a hurry
- Make his decision simple
  - give him ready arguments for your paper
  - do not expect him to infer them from your text
- Specify what the *novelty* and *importance* is
- Specify *why* your method is better

# Conciseness

- Omit UNIMPORTANT details
- Choose shortest way to say THE SAME  
the same = not sacrifice clarity
- When I want readers to understand me,  
I use fewer words / letters
  - I re-write my text in several passes
- When I want readers NOT to understand, more

# Don't

- Given our considerations laid out above in the current section, one can easily conclude that in fact the value of the variable  $a$  can be regarded as not showing any difference as compared with the corresponding value of the variable  $b$

# Do

- Thus,  $a = b$

*Exactly the same meaning*

# Good vs. bad papers

- Good papers:
  - solid,
  - easy to follow
  - good state of the art overview
  - detailed report of experiments
- Bad papers:
  - bad English
  - inconclusive
  - hard to follow, important details omitted

# Focused

- Do not boast of your knowledge
- Do not fill the pages
- Tell the story!
  - When you return from a journey, you tell your friends about your journey, not your knowledge of geography
- Start writing with important things. Don't omit important details. You will need more pages, not need to fill pages!

# Important things first

- **Title:** main idea. Write 10, leave 3, choose 1
- **Abstract:** plan / what you want to tell the world

# Important things first

- Title
- Abstract
  
- **Method:** all *relevant* details needed to reproduce



# Important things first

- Title
- Abstract
  
- Method
- **Experimental results:** compare with SoA

# Important things first

- Title
- Abstract
  
- Method
- Experimental results
- **Discussion:** advantages / disadvantages; why

# Important things first

- Title
- Abstract
  
- Method
- Experimental results
- Discussion
- **Conclusion:** abstract for expert; future work

# Important things first

- Title
- Abstract
- **State of the art:** who tried to solve this problem
- Method
- Experimental results
- Discussion
- Conclusion

# Important things first

- Title
- Abstract
- **Intro:** importance, novelty, overview, ToC
- State of the art
- Method
- Experimental results
- Discussion
- Conclusion

# Important things first

- **Title:** does it correspond to the text?
- **Abstract:** does it reflect the main ideas?
- Intro
- State of the art
- Method
- Experimental results
- Discussion
- Conclusion

# Important things first

7. Title

8. Abstract

6. Intro

5. State of the art

1. Method

2. Experimental results

3. Discussion

4. Conclusion





# Where to send

- Publish in journal of *applications*, not of *method*
- Fast answer: look at papers
- Journal that publishes more papers
- Look at other papers in this specific journal: how they look, how they reason, how important things communicate

# Explanation

Make others know  
what you know

Before PhD      you      From PhD



# Don't



# Do



# Pyramid of knowledge

Top: not much info

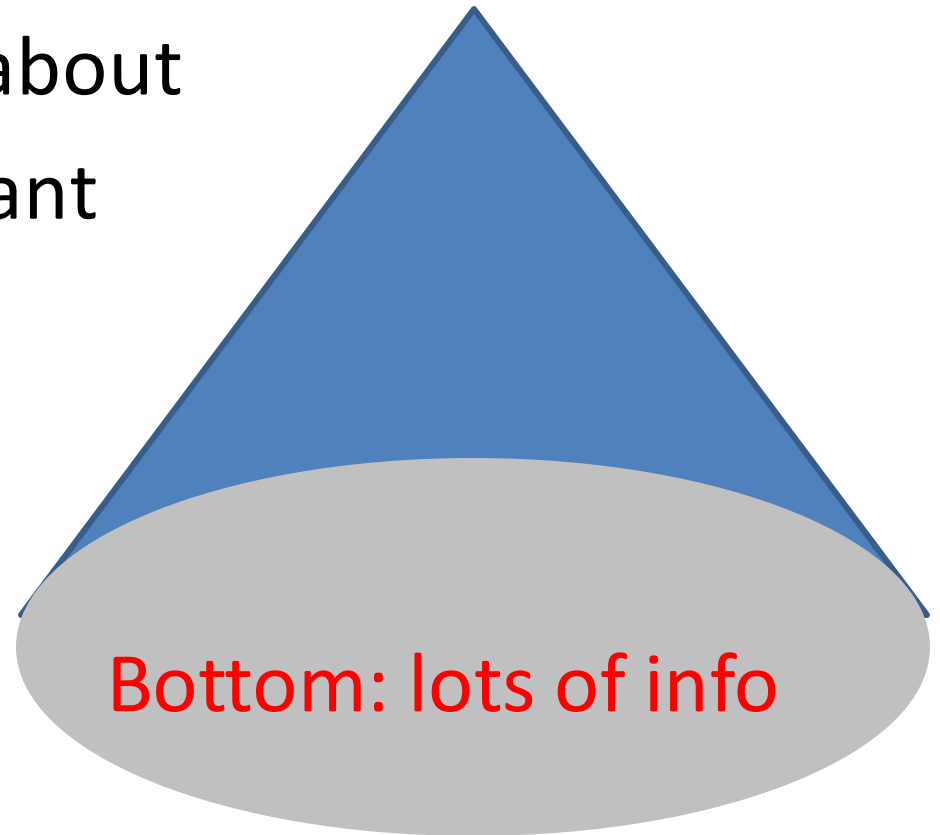
What it is all about

Why is it important

General idea

Structure

Technical details



Bottom: lots of info

# Pyramid of knowledge

One cannot  
understand  
the bottom  
without understanding  
the top



# Two situation of explanation

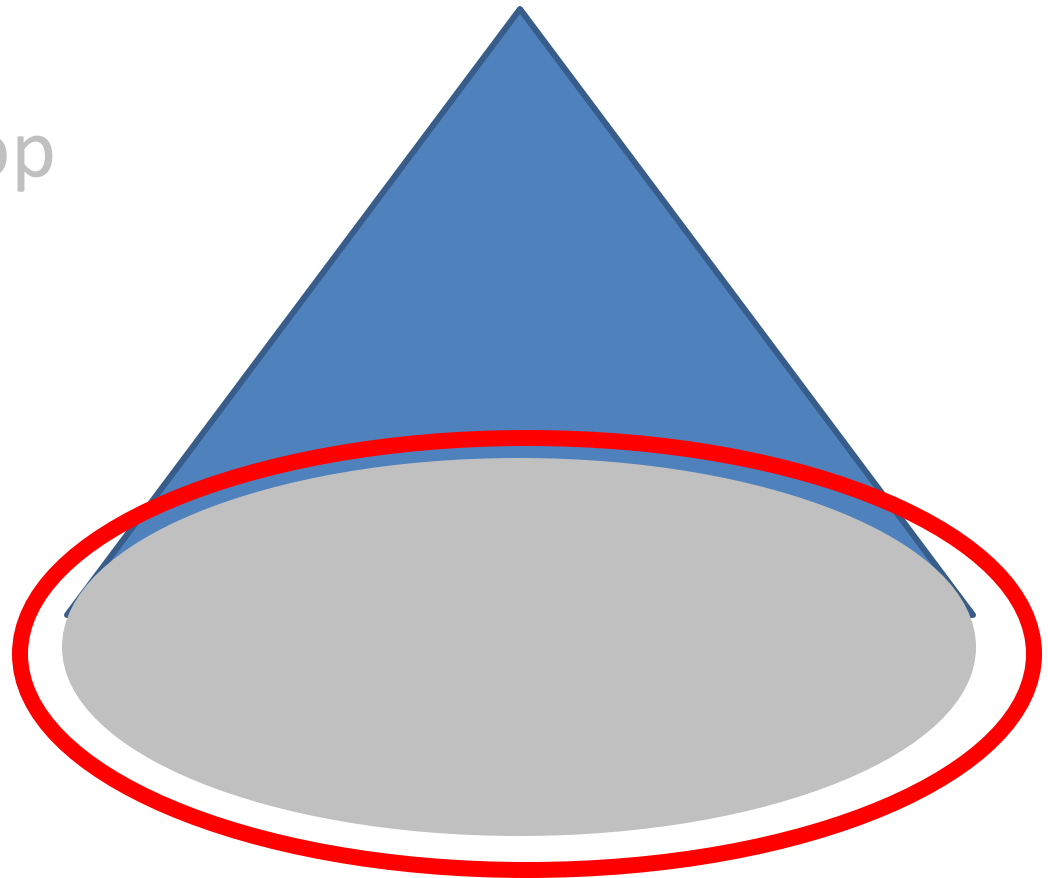
- Explain to one who already knows
  - Their goal: check if you know  
They are interested in you
  - Your goal: demonstrate that you know
- Explain to one who does not know
  - Their goal: to know  
They don't care about you and if you know
  - Your goal: pass the knowledge,  
make that they know

# Optimal strategy in high school

One cannot understand  
the bottom without  
understanding the top

Cover the bottom.

This covers  
the top, too

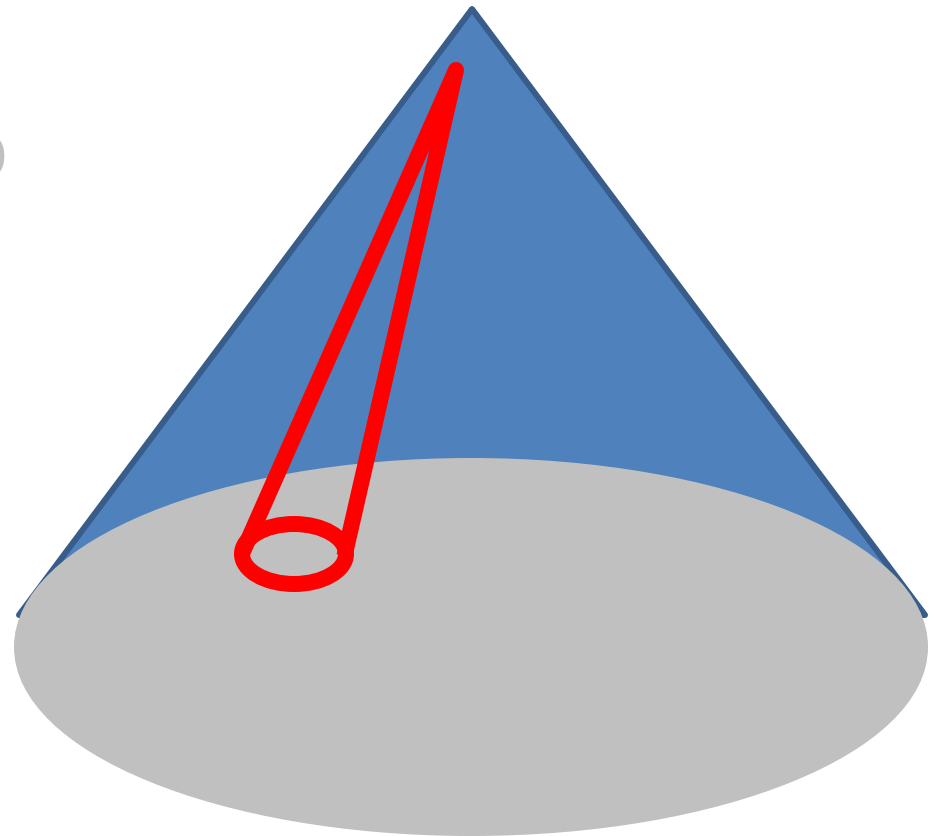




# Optimal strategy in high school

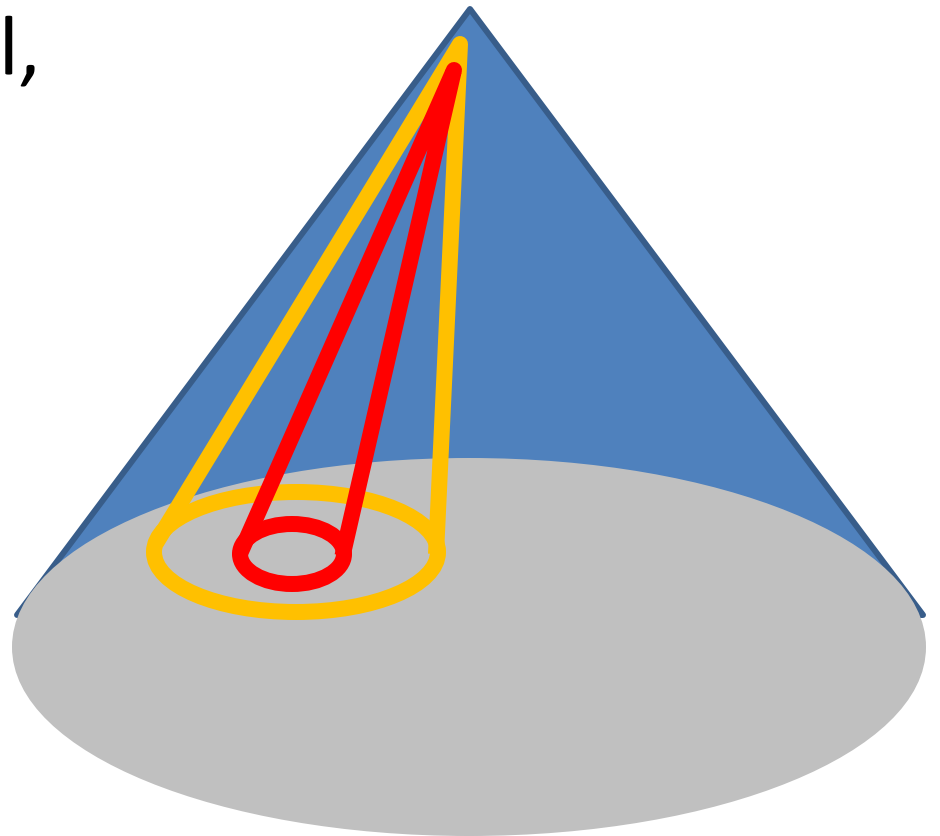
One cannot understand the bottom without understanding the top

When you mention a small technical detail, it covers things above



# Optimal strategy in high school

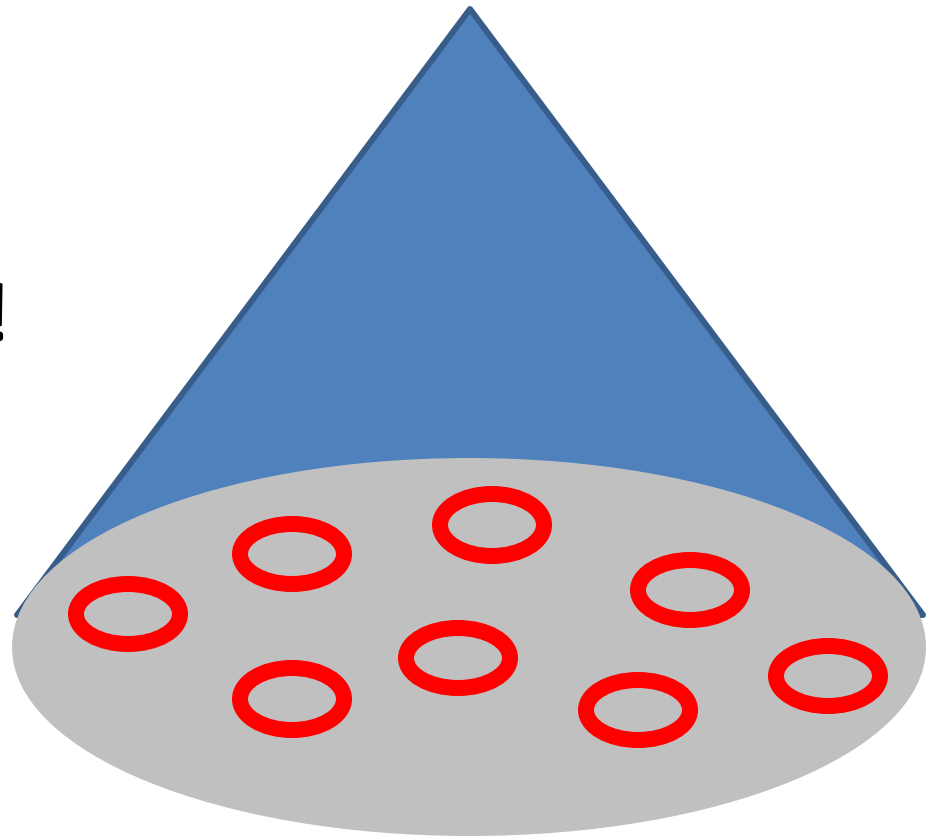
When you mention  
a small technical detail,  
this covers  
things around



# Optimal strategy in high school

Mention random things  
from the bottom  
here and there.  
This covers the whole!

This way, you  
demonstrate  
that *you* know



# Explaining to your professor

- Android's current release is 4.4.2
- Based on the Linux kernel version 3.4.10
- File system based on i-nodes
- 32-bit ARMv7 architecture
- Supports OpenGL ES 1.1, 2.0 and 3.0
- Android uses a software stack
- uses Bionic in place of a standard C library

# Explaining to your professor

- Android's
- Based on
- File system
- 32-bit ARM
- Supports
- Android u
- uses Bionic in place of a standard C library



3.4.10

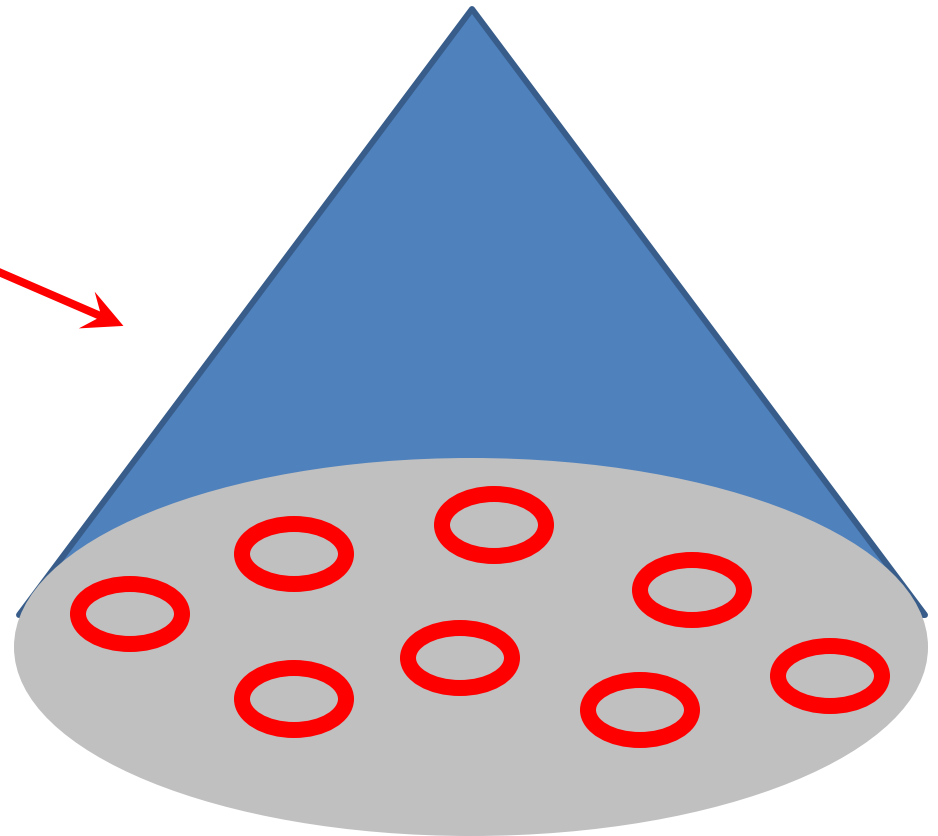
.0

**Look mom, I know what Android is!**

# Don't

So, students write papers  
like this

Fails horribly  
when explaining to  
one who does not  
know in advance!



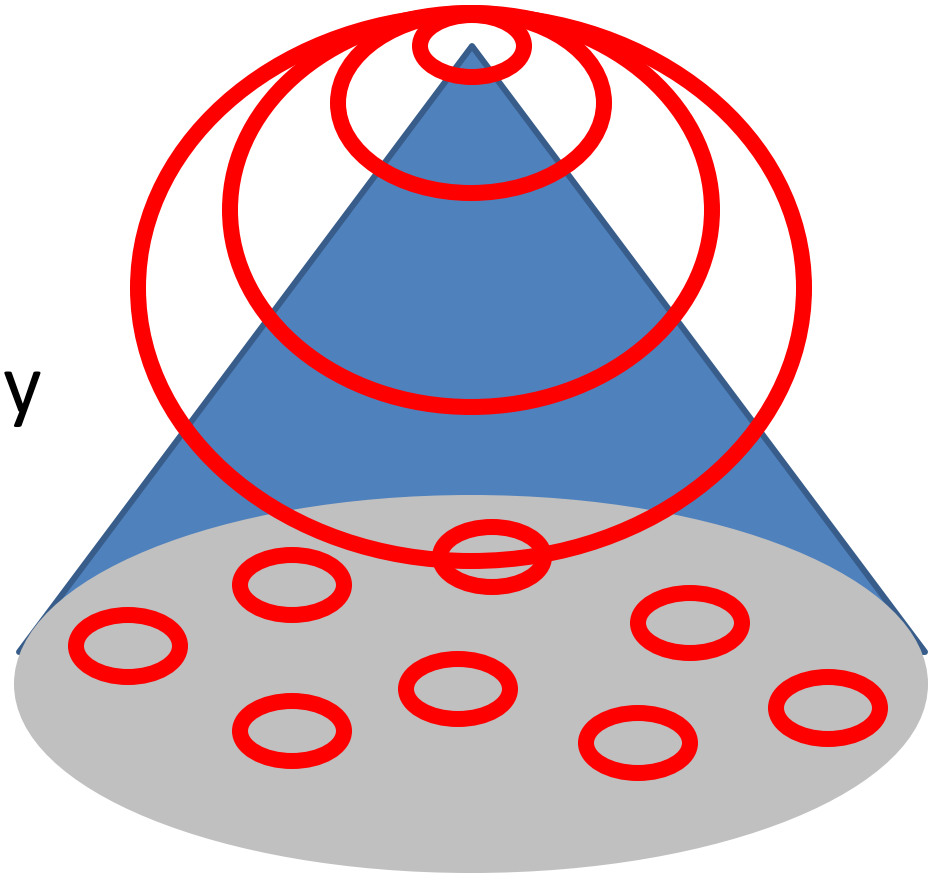
# Optimal strategy: PhD

Start from the top.

Tell the story.

If you have time only  
to explain the top, they  
get *something*.

Explain from the  
very beginning

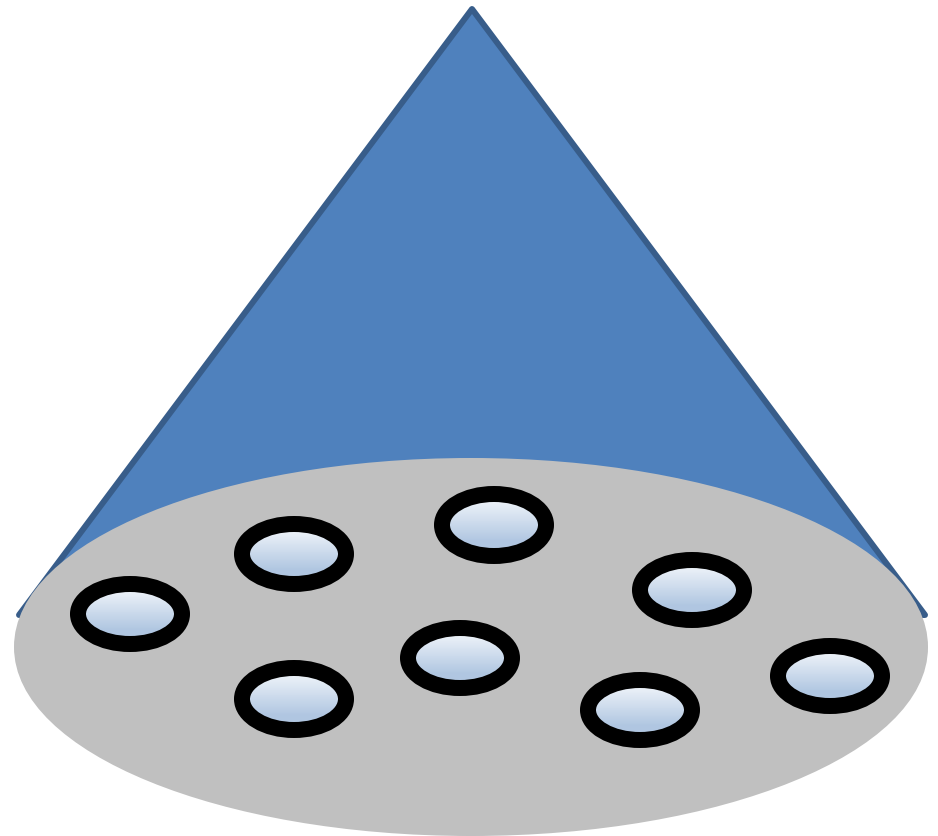


# Optimal strategy: PhD

Start from the top.

If you start from  
bottom, they get  
*nothing*

Not even these  
details





# Explaining to your professor

- Android's current release is 4.4.2
- Based on the Linux kernel version 3.4.10
- File system based on i-nodes
- 32-bit ARMv7 architecture
- Supports OpenGL ES 1.1, 2.0 and 3.0
- Android uses a software stack
- uses Bionic in place of a standard C library

# Explaining to your professor

- Android's
- Based on
- File system
- 32-bit ARM
- Supports
- Android u
- uses Bionic in place of a standard C library



3.4.10

.0

**Look mom, \*I\* know what Android is!**

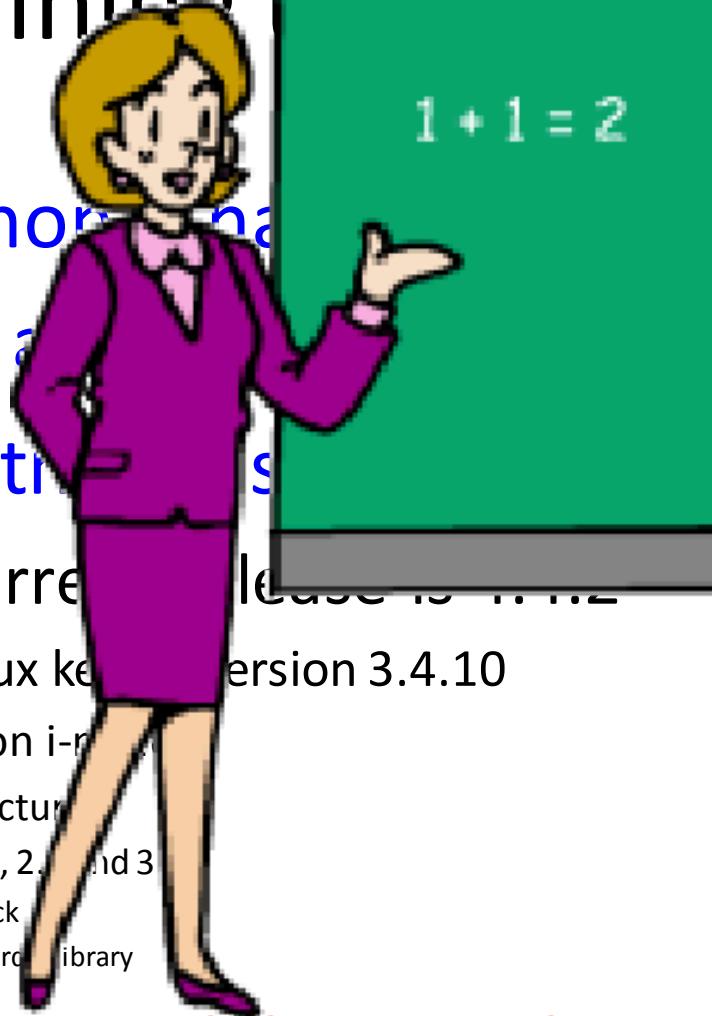
# Explaining to your grandma

- Modern phones have complex behavior
- They need an operating system
- Android is the most popular such system
- Android's current release is 4.4.2
- Based on the Linux kernel version 3.4.10
- File system based on i-nodes
- 32-bit ARMv7 architecture
- Supports OpenGL ES 1.1, 2.0 and 3.0
- Android uses a software stack
- uses Bionic in place of a standard C library

**Now SHE will know what Android is**

# Explaining to your grandma

- Modern phone has different behavior
- They need a different system
- Android is the best system
- Android's current release is 4.1.2
- Based on the Linux kernel version 3.4.10
- File system based on i-n
- 32-bit ARMv7 architecture
- Supports OpenGL ES 1.1, 2.0 and 3.0
- Android uses a software stack
- uses Bionic in place of a standard library

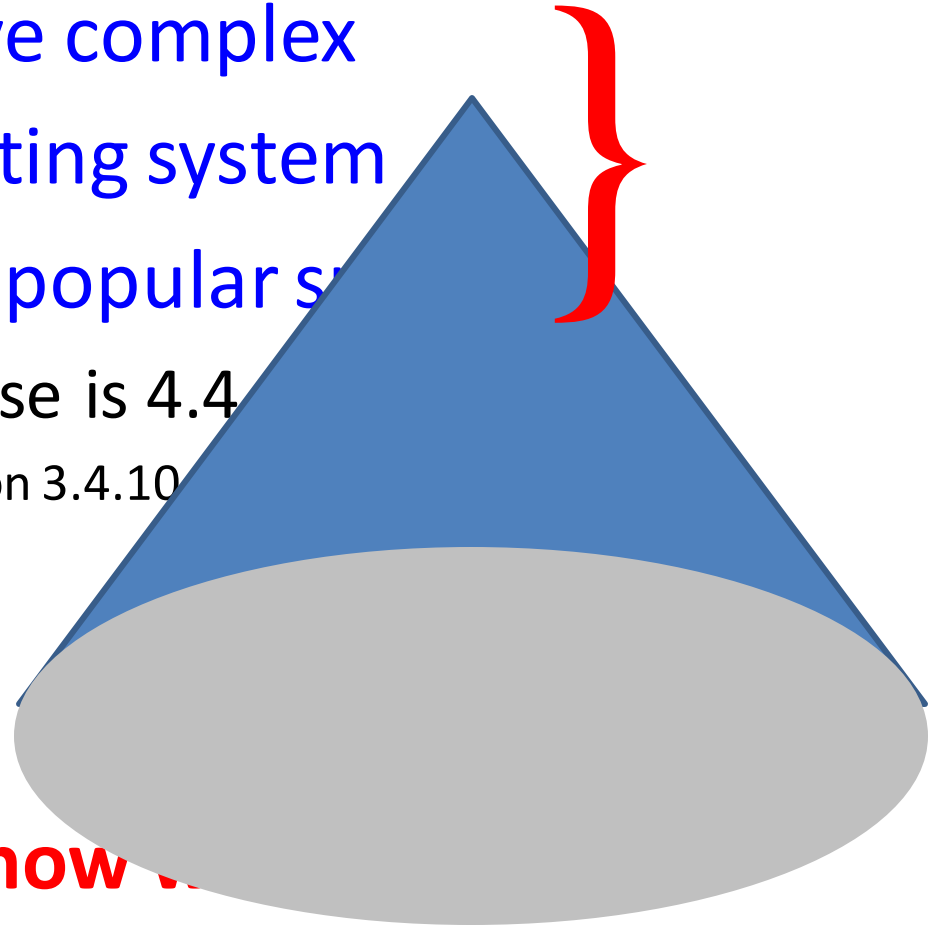


**Now SHE will know what Android is**

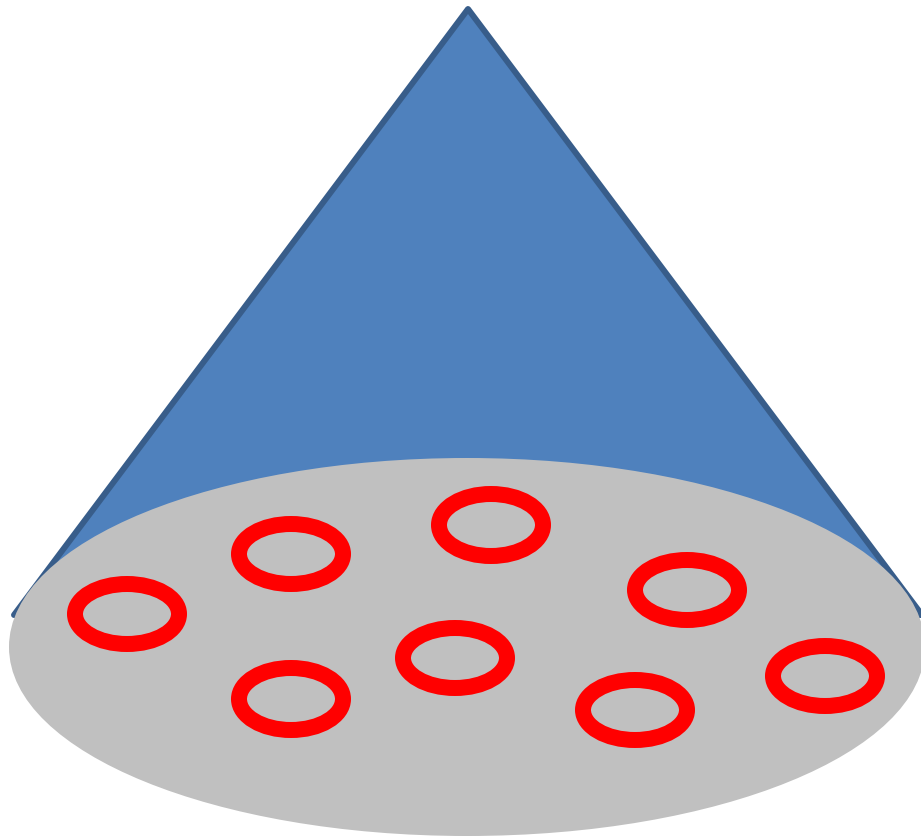
# Explaining to your grandma

- Modern phones have complex
- They need an operating system
- Android is the most popular system
- Android's current release is 4.4
- Based on the Linux kernel version 3.4.10
- File system based on i-nodes
- 32-bit ARMv7 architecture
- Supports OpenGL ES 1.1, 2.0 and 3.0
- Android uses a software stack
- uses Bionic in place of a standard C library

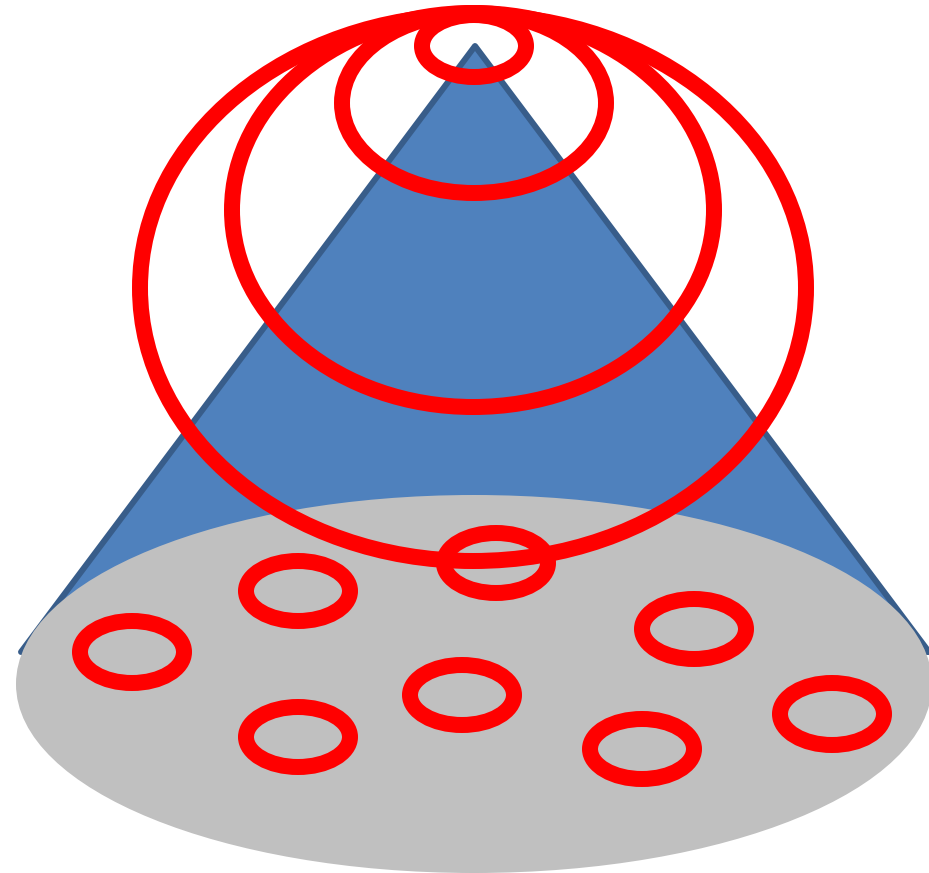
**Now SHE will know**



# Don't



# Do



Don't build systems

Write working proofs

# Industry integration



# Science analysis





# Industry



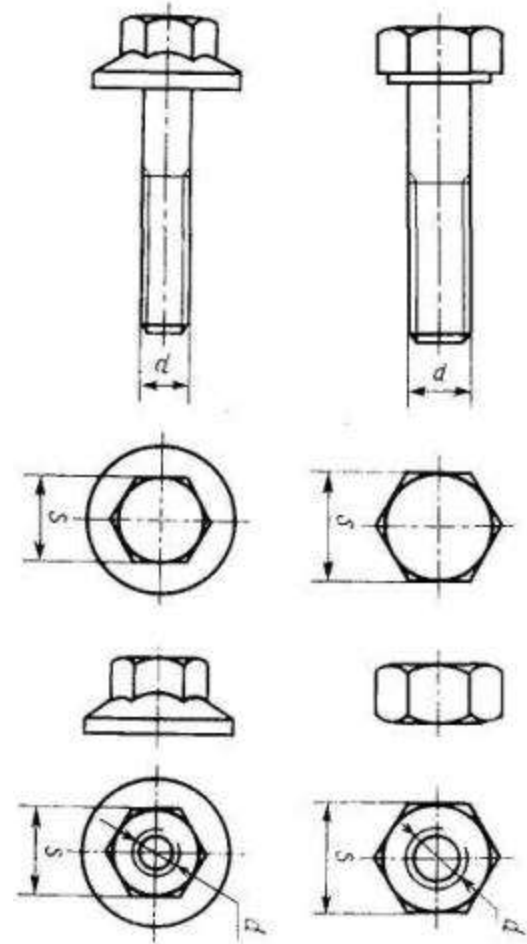
# Science



# Industry



# Science



# Industry

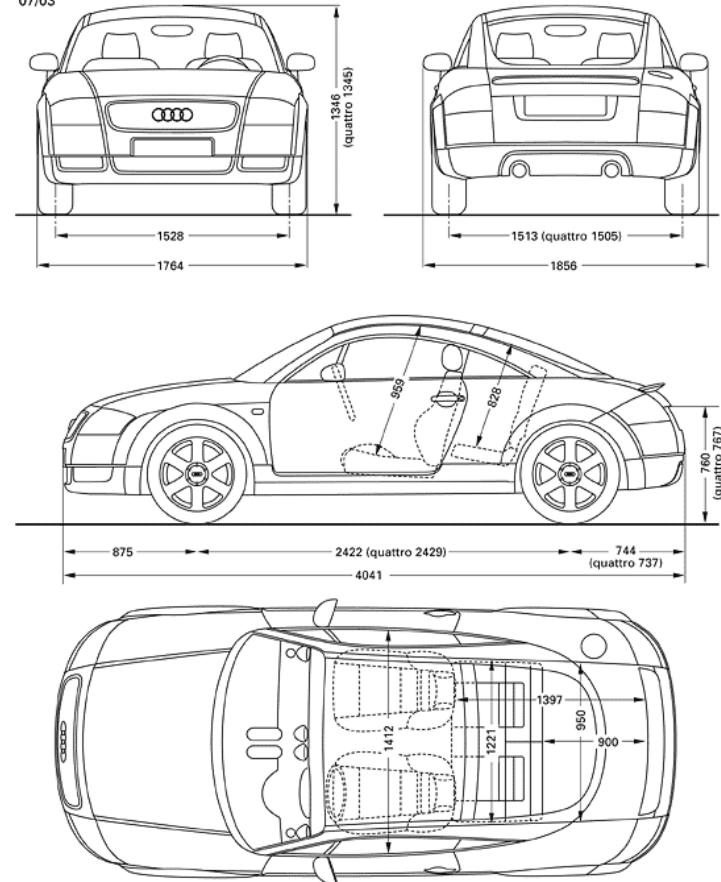
# Science



## Audi TT Coupé

Abmessungen  
measurements

07/03



# Are you a good programmer?

irrelevant



# Goal of a program: prove

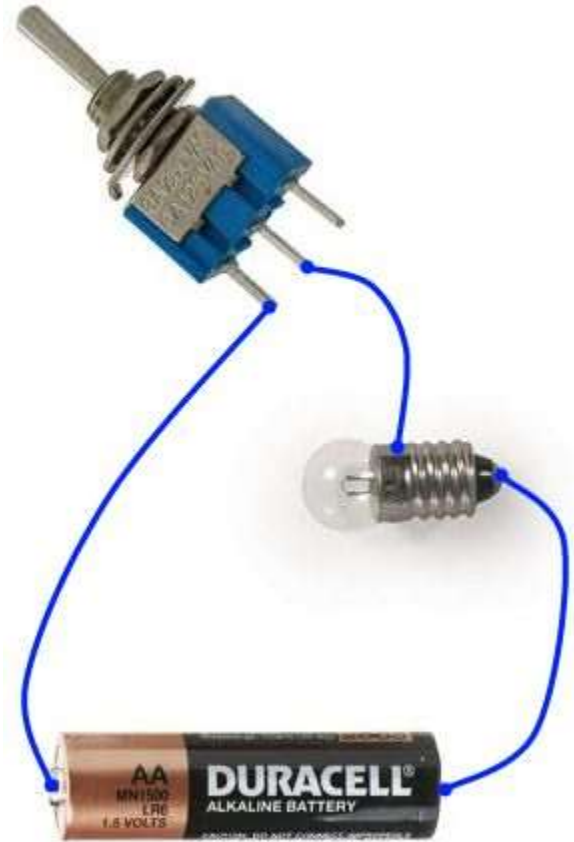
- Prove the idea,  
not that you are great programmer
- No windows, no complexities, no fancy libraries and effects
- Simplest algorithm possible
  - otherwise you can't publish it
- If your goal is to prove  $X$ , only prove  $X$ , don't build a system around  $X$



# Don't



# Do

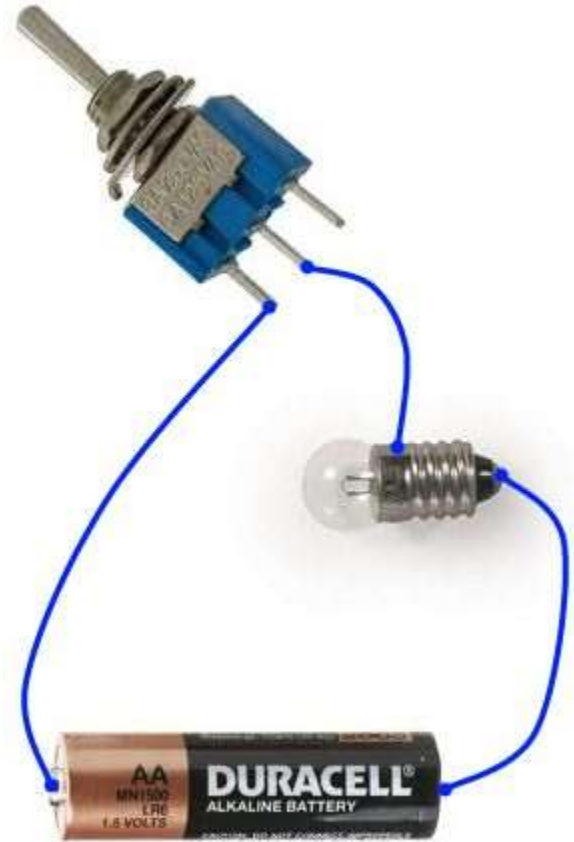


# In PhD

Don't

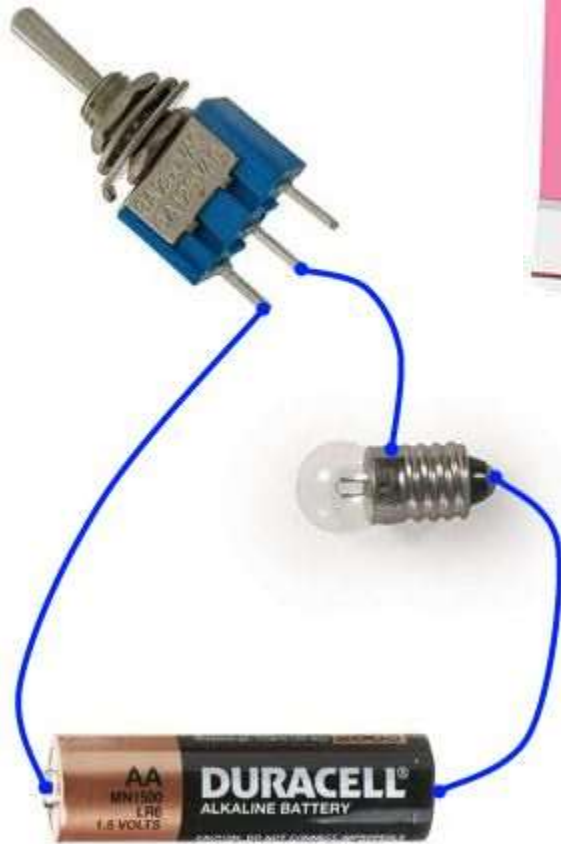


Do



# In BSc

Don't



Do





# Goal of a program: prove

- **Building a system does not earn you PhD, proving an idea does**
- Complex system makes proof opaque
  - Magic, not science
  - Screen between the observer and phenomenon

# Complex system: difficult to report

- Competitions of systems
  - You use 45 features
  - You win the competition
  - You become champion, but not PhD
- Pizza effect:  
The more you add, the better the results
  - Can you publish this?
  - Good new system. No new knowledge

# Invent ideas, not build systems

- **Idea:** what you put in the title of a paper
- A theorem
  - and the paper is the proof

- If you built a complex system

The *theorem* is:

the more effort I spend the better the result

- not novel
- not science (yes, industry)

# Don't build systems

- You are not a programmer
  - Your system will not be used
  - Algorithm is 10% of the effort, 90% to product
- Does not prove the idea
- Wastes your time
- You are left with no system and no proof
  - “I worked a lot” does not prove the theorem
  - “I built a huge system”, neither

# Build toolboxes

- Research tools
  - highly configurable
  - highly interoperable in simple ways (files!)
  - you did not try every combination of options, let others do
- Open source
- Simple
  - otherwise no use of open source
- Service to the community

# I don't say “do not program”

- Minimal proof: focus on **simplicity of code**
  - human-readable code
  - proof of a theorem, written in Java
- System: focus on **usability**
  - pizza effect: the more the better
  - code does not matter
- Toolbox: focus on **flexibility** + simplicity
  - framework to write proofs of theorems in Java

# PhD

- Building a system does not earn you PhD
- Building a proof does
  - to prove, you need to **program**
  - lot of programming already
  - do this first, and do it well
  - you will not have time for a *system*
- Building a toolbox does not earn you PhD
  - but, your service to the community
  - indirect contribution in advancing knowledge

# Paper

- Whenever you find yourself writing the word “system” in your paper, throw away your paper and sell your system to the industry
- Or, throw away your “system” and re-think what you are doing in terms of research questions



# Paper

- By definition, system is a combination of many things
  - not analyzable
  - not reproducible
  - not re-usable
- Your system works well? Why? No idea.
- The reader does not learn anything from this
- What are the lessons learnt from your work?

# Paper

- Instead, describe your method
  - analyzable: steps
    - analyze impact of each step
    - justify each step: science is simplicity
  - reproducible: describe how
  - re-usable: can be used on in others' systems
- Don't boast of your "system," make readers learn something useful in their own work

# Research

Think in terms of research questions,  
not tasks or systems

# Research questions

- Goal of science is understanding, not solving tasks
- Research should answer a question
- Formulate what your research question is, and what type of answer you expect
  - “build a system” is not an *answer*
- If you cannot, not a good topic

# List of stupid questions

- When reading, thinking, working, write down questions / ideas
  - each questions, a future paper
- First write down, then think
- Explain, in one paragraph
- Questions grow in trees of questions

# Choice of topic

- Research topic is love
- Choosing research topic is marriage
  - so difficult and frustrating
  - only can be endured with great love!
- Very personal choice
  - We don't expect father to tell us who to love
  - Don't expect your supervisor to give you the topic
- People work best on what they chose

# Choice of topic

Like choosing a wife!

Get to know many

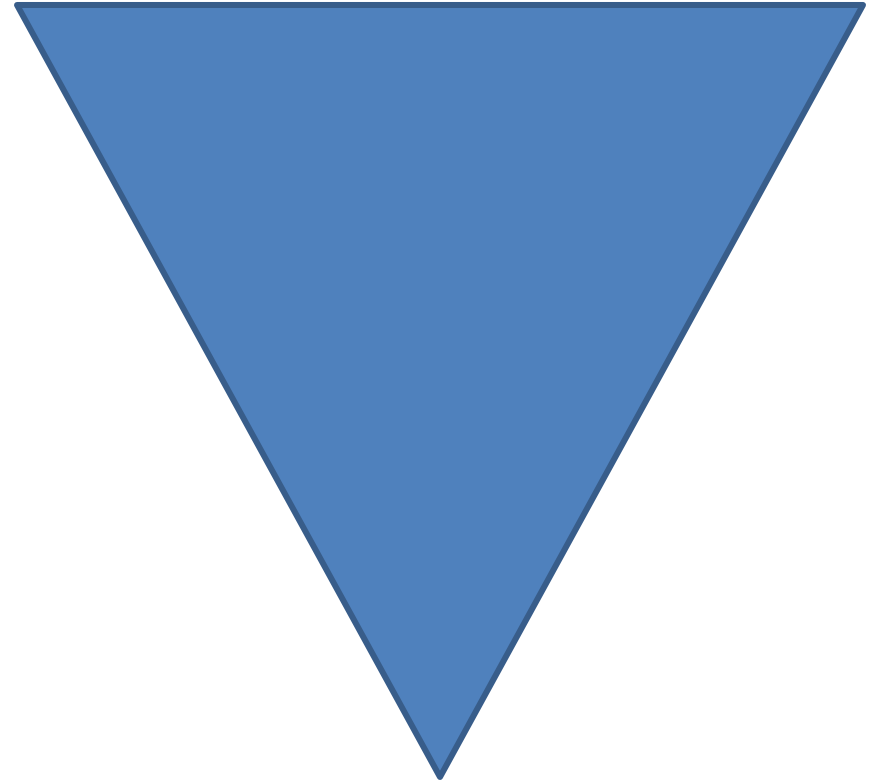
Narrow the search

Choose a few

Choose one

Change if does not work

Read much and consider many, not the first



# Choice of topic

Like choosing a wife!

Read abstracts of 100

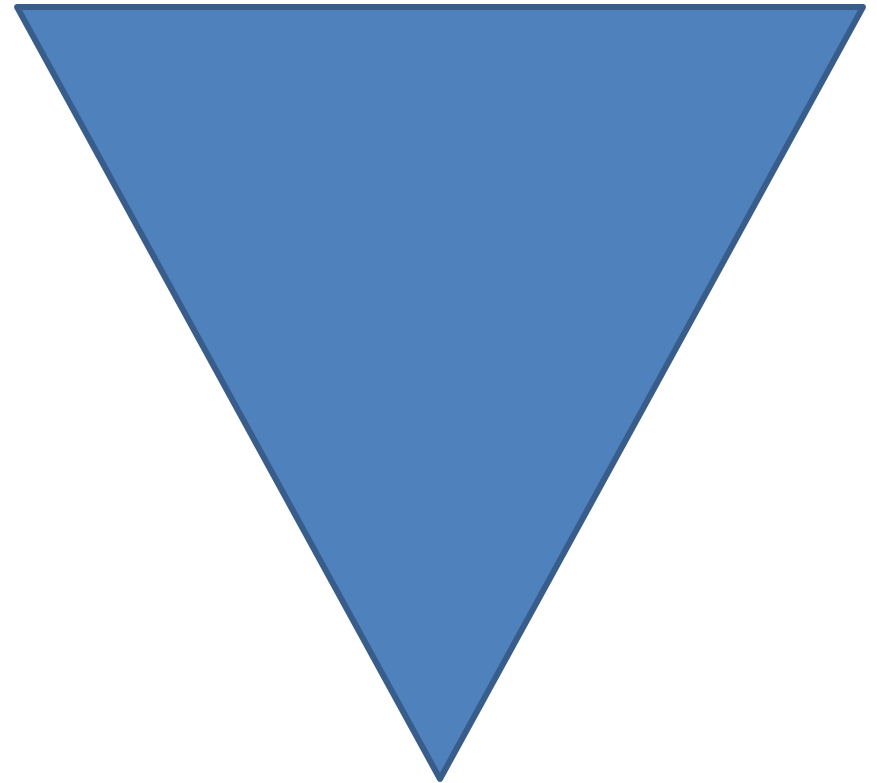
Read intro + concl of 50

Read and understand 15

Choose one

Change if does not work

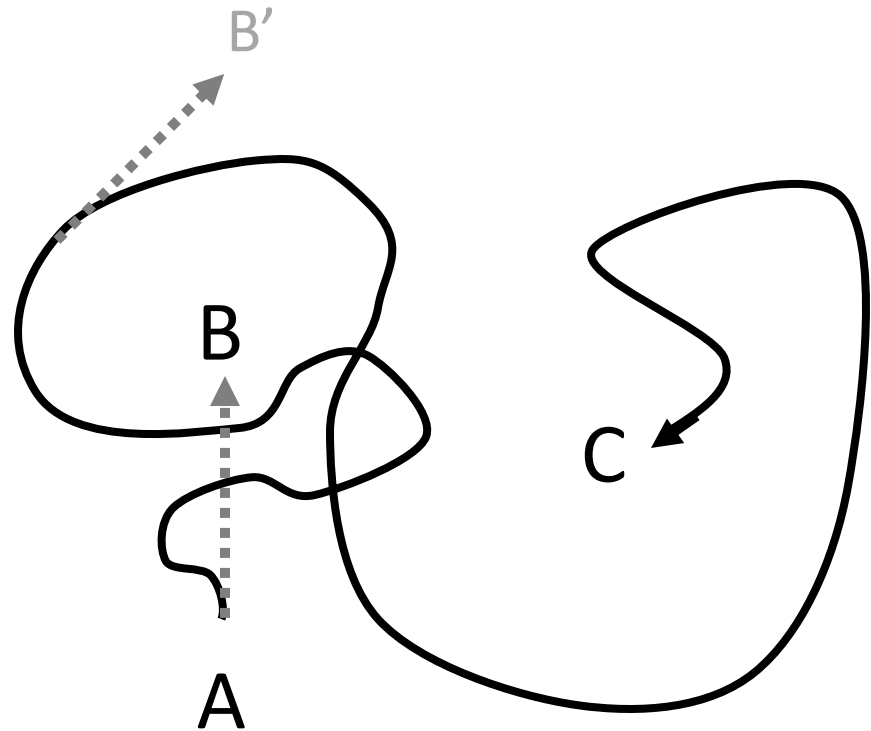
Read much and consider many, not the first





# Research trajectory

I like planning!  
(just kidding)



This is normal. This is re-SEARCH

# Five questions

1. Has this been done?
2. Can this be done?
3. Anybody needs this to be done?
  - many people tried, nobody achieved
4. Am I the smartest person in the world?
5. What is my special circumstance?

**Exploit and reinforce your difference. Don't compete where you have no advantage**

# Feasibility

- Feasibility, more important than greatness
- No such thing as too small topic
  - if done well
  - explore all options, report all details
  - one answered question causes ten new

# Don't



2016

# Do

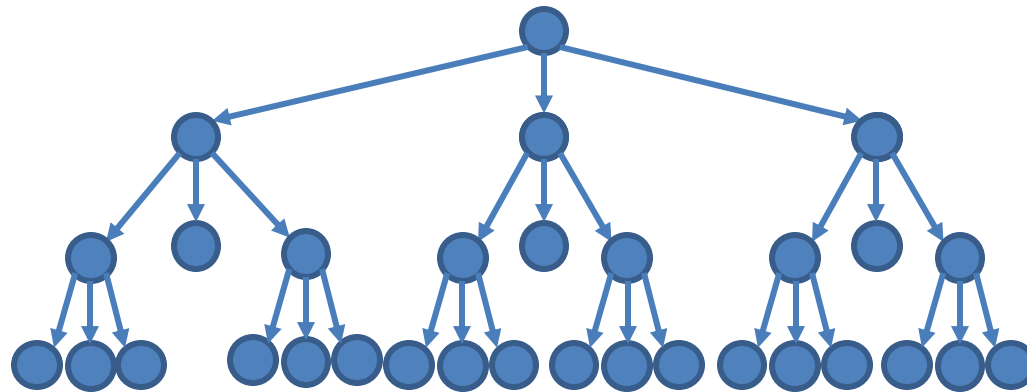


(c) Alexander Gelbukh - [www.Gelbukh.com](http://www.Gelbukh.com)

100

# Develop incrementally

- Tree of goals:  
*to do this,*  
*I need that and that*

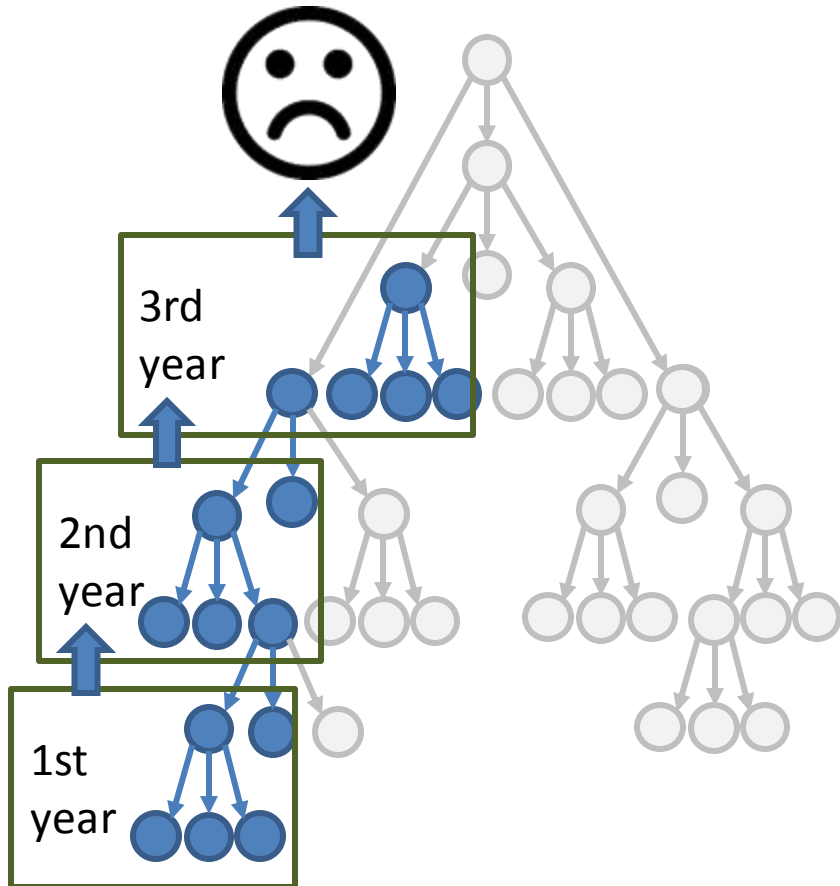


# Develop incrementally

Have something working at any moment

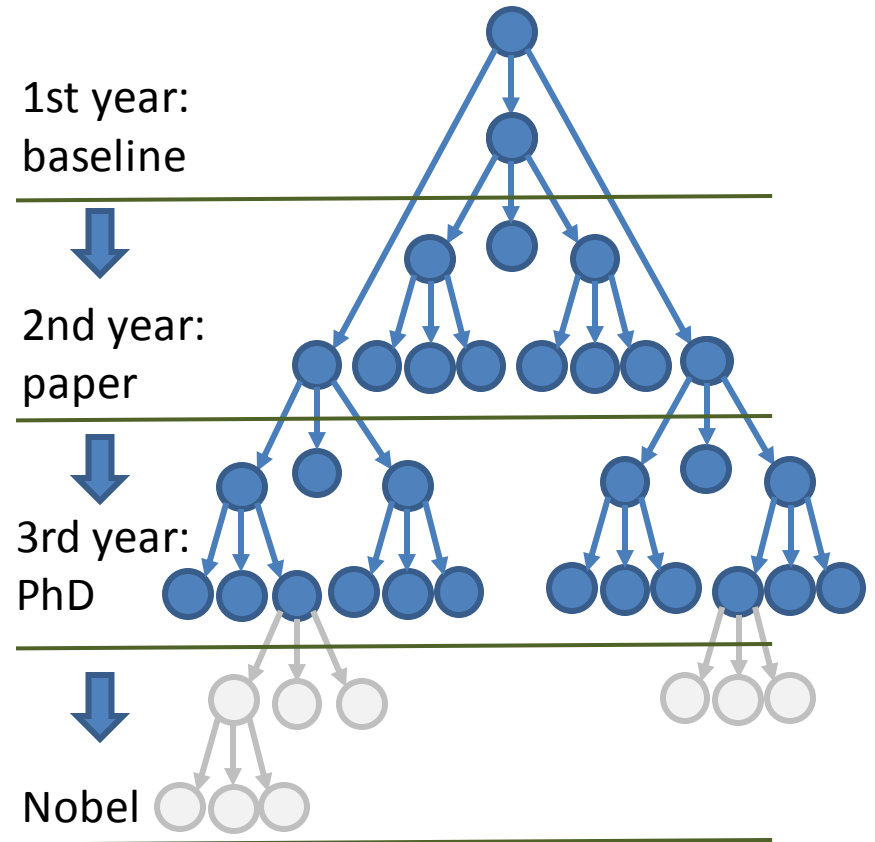
- Implement the simplest method (baseline?)
- For each part:
  - start from trivial implementation / placeholder
  - decide what part needs improvement most
  - improve gradually, add complexity step by step
- Keep whole thing functional at each stage
- When time is over (oops!), you have *something*

# Don't



integration

# Do



incremental improvement

Reading

Generalize



# What to note when reading

- Chef in a restaurant
  - how dishes were prepared, to learn
- Why do you like the paper
  - note the tricks the author used to make the paper easy to read
- What do you not like in the paper
  - to avoid
  - reviewing papers is very useful

# What to note when reading

- Tree of problems and solutions
  - authors do not always make it explicit
  - what was the main problem addressed by the paper?
  - what was its solution?
  - what were the problems of this solution?
  - what were their solutions?
  - make it explicit

# Understand = simplify

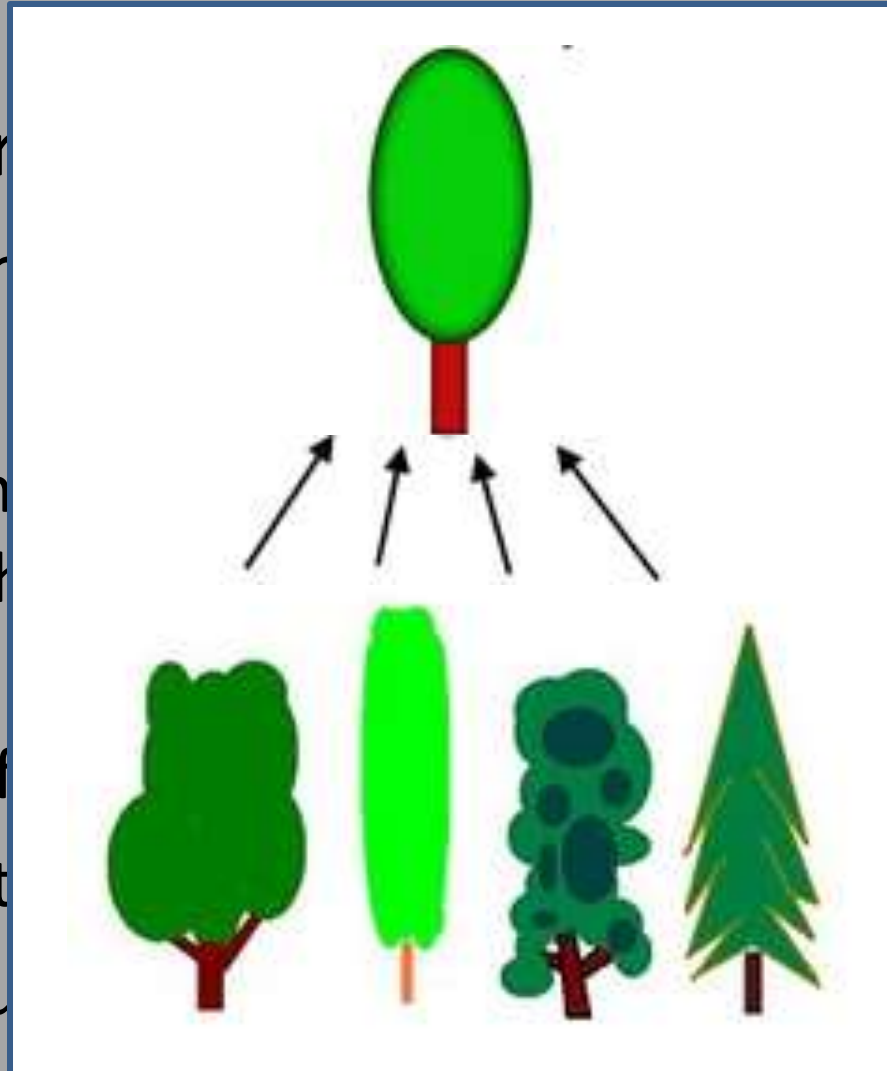
- Re-formulate the main idea in short words understandable for your supervisor and colleagues
  - This is what Partha does, but using constituency tree
  - Like SVM but not linear
- Helps you / your colleagues to understand
- Simplifies discussions

# Look for alternatives

- What were options and parameters?
- Author only tried some options, what were others?
  - the author might or might not realize that there were others!
  - you can do it! Each option they did not try, is a paper of yours
  - parameters
  - author used  $a + b$ , oops:  $ka + (1 - k)b$

# Look for alternatives

- What were
- Author or others?
  - the author were other
  - you can paper of
  - parameter
  - author u

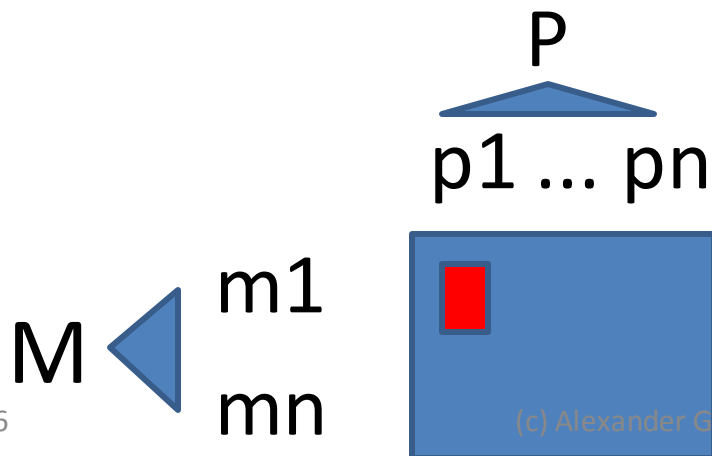


?  
at were  
that there  
t try, is a

# Generalize

The author solved problem  $p$  with method  $m$

- Is  $p$  a particular case of some type  $P$ ?
- Is  $m$  a particular case of some type  $M$ ?
- If so, many problems of type  $P$  can be solved with many methods of type  $M$



# Generalize

At any point of the problem-solution tree, ask yourself three questions:

1. What other method can I use to solve  $p$ ?
  - if author writes about  $p$ , solving it is important
2. What other tasks can I solve with  $m$ ?
  - author worried only about  $p$
3. What similar tasks can I solve with similar methods?

# Conclusions

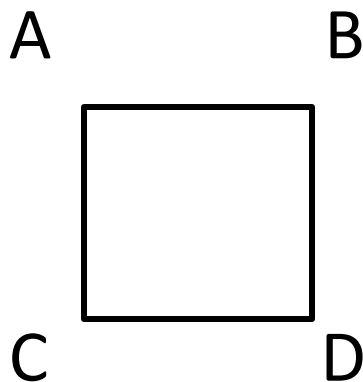
Checklist



- Advancing knowledge of humankind?
- Or having fun for yourself?



- Have you made every effort to keep it simple?
- Or making scientific-looking appearance?



*We consider a simplicial complex that consists of four one-dimensional geometric shapes of equal measure whose zero-dimensional cells are pairwise identified, along with the corresponding immersion in standard  $\mathbf{R}^2$*

- Your work is great.
- Your PAPER, which one?



- Your paper: masterpiece?
- Or formality?

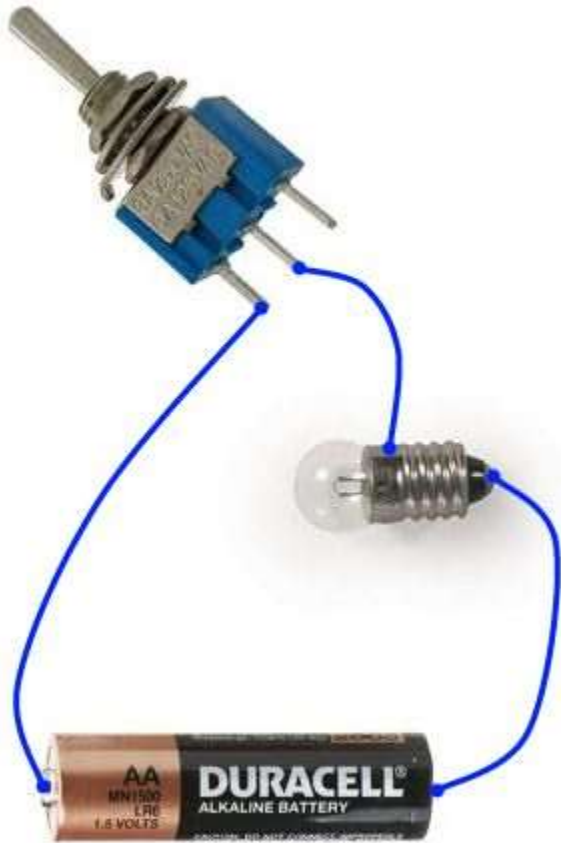




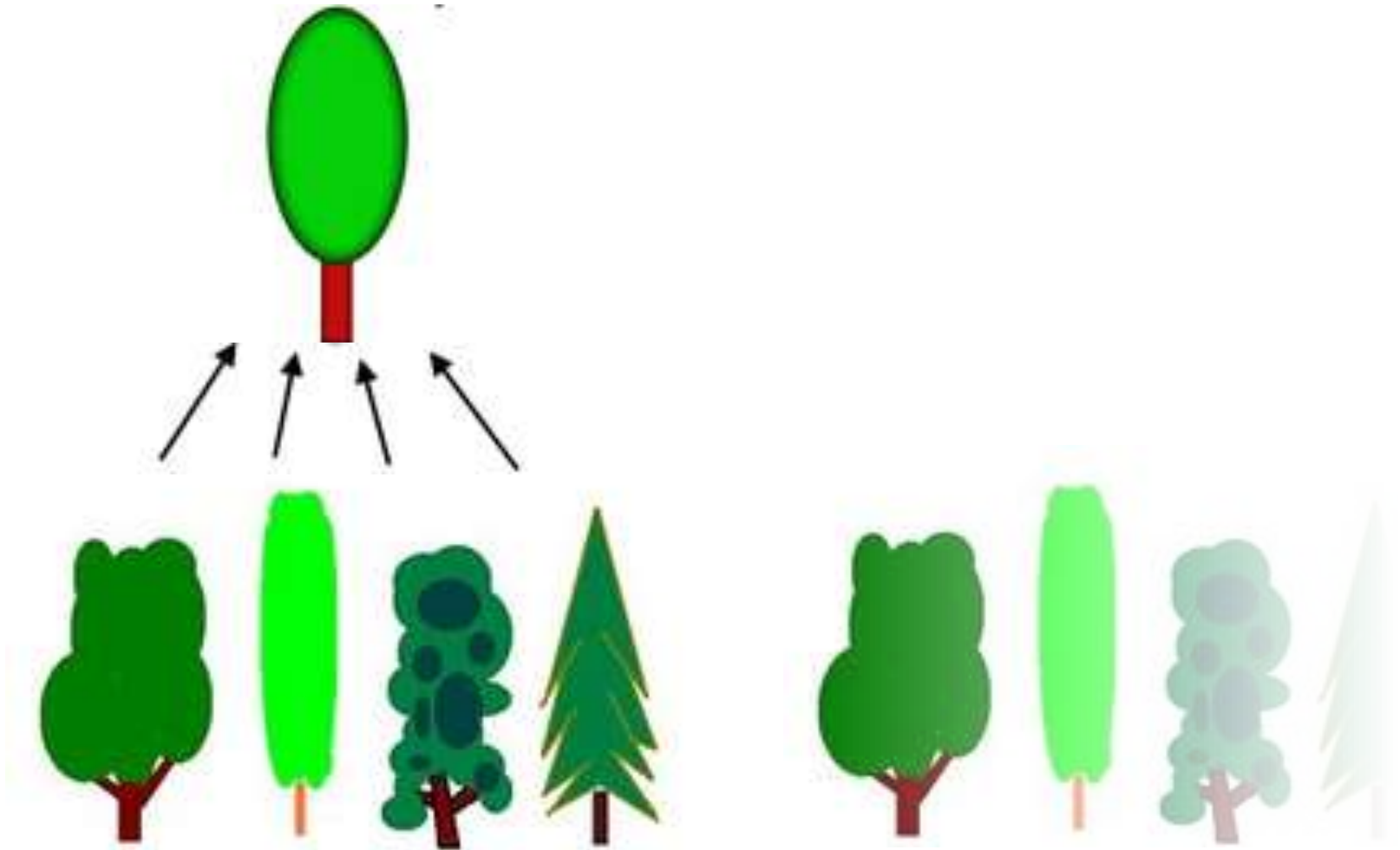
- Can you see the differences that matter in the reviewer's culture?



- You spent 3 years in PhD
- Did others LEARN from what you achieved?



- Know how to cultivate trees?
- Or how to cultivate apple trees?



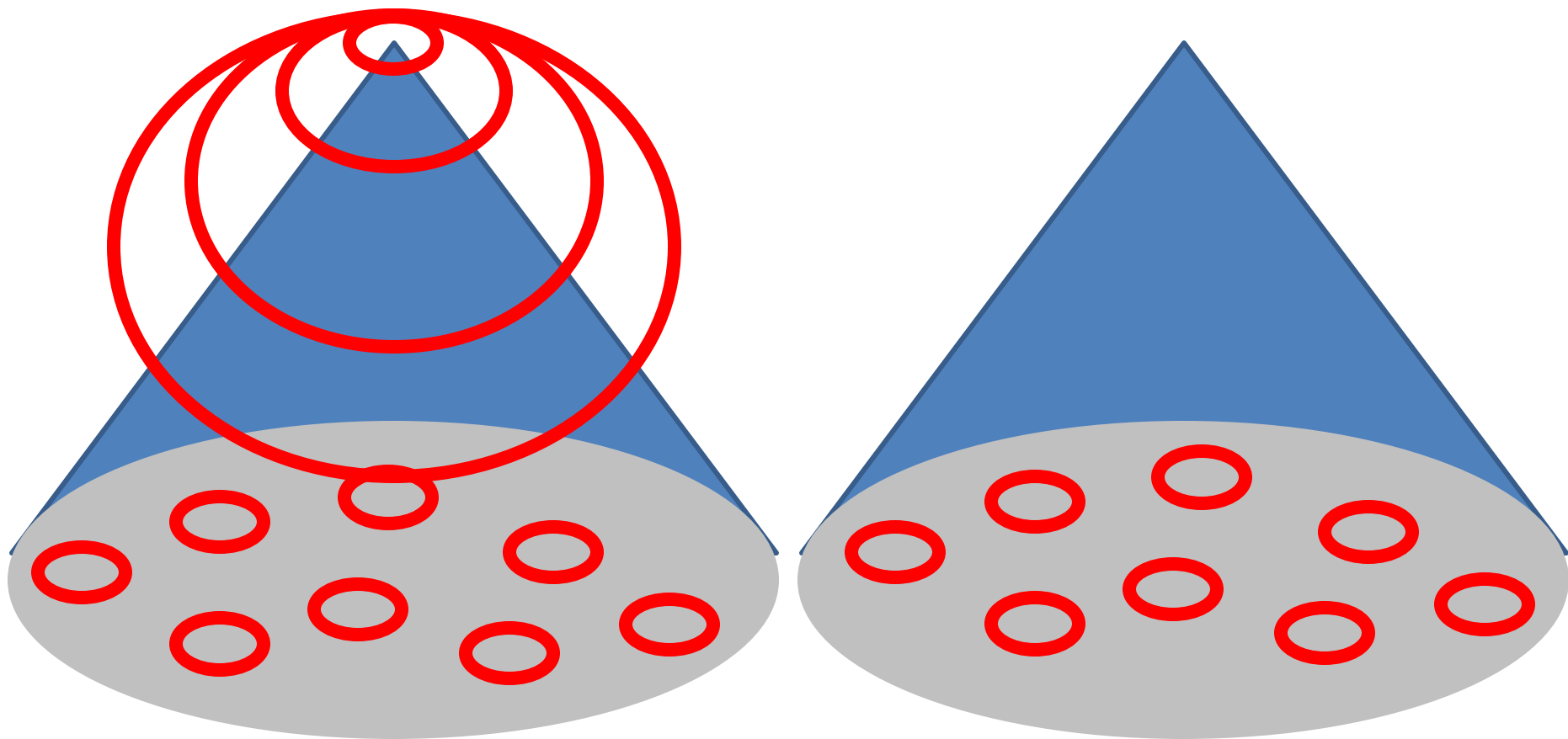


Explaining something useful?  
Or that you are good programmer?

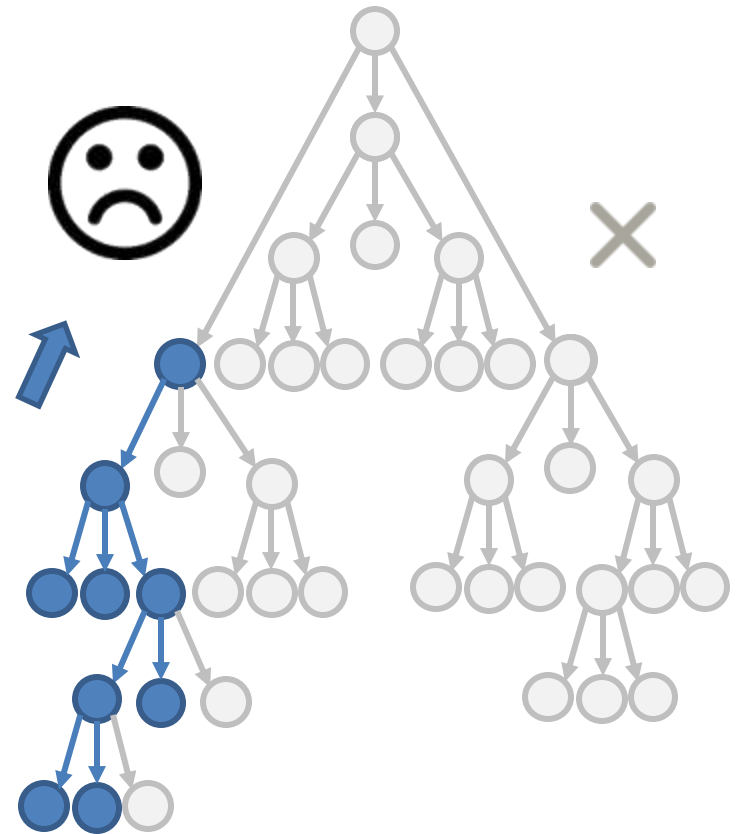
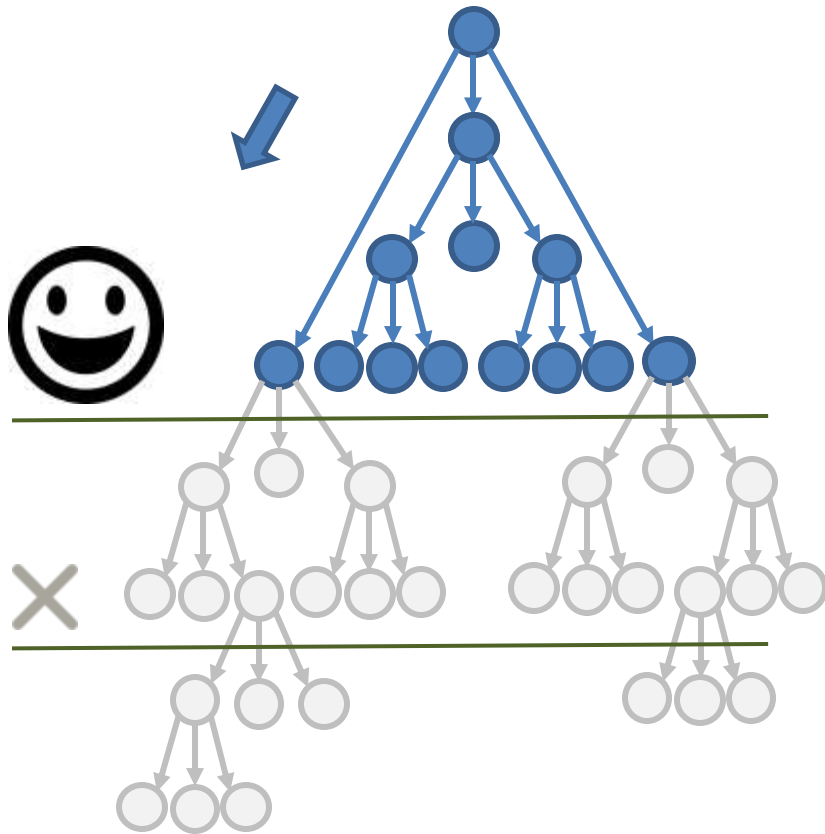




- You know what you want to say
- Will others understand you?



- Oops... time over! Tomorrow's the defense.
- Ready with *something*, or have first module?



Gave a definitive answer to a (small) research question?  
Or had just time to almost get ready to start?





Thank you!

[nlp.cic.ipn.mx](http://nlp.cic.ipn.mx)