



PRESENTING A
PROFESSIONAL

PROFESSIONAL
PRESENTATION

By Michael Goldwasser



OUR GOAL

- ❖ What makes an effective (technical) presentation?
- ❖ Perhaps some of you had had public speaking class?
- ❖ Perhaps you have made other technical presentations?
- ❖ Professional meetings as intern?
- ❖ No experience? We're here to help



GENERAL OVERVIEW

- ❖ Be Professional
- ❖ Be Organized
- ❖ Manage Your Time
- ❖ Typical outline:
 - Project Overview
 - Dig Deeper
 - Timeline, Responsibilities
 - Demo(?)
 - Conclusions, Questions

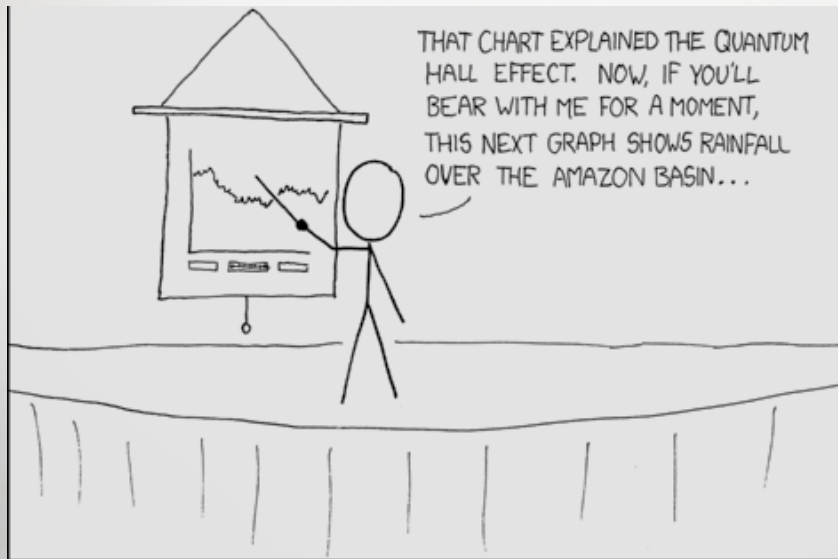


BE PROFESSIONAL

- ❖ Be Prompt (in fact, be early)
- ❖ Attire: Don't need to be fancy (but don't embarrass yourself)
- ❖ Consider Visual Appeal of Presentation Materials

- ❖ Test your slides/technology/computer in advance
(if you need advanced access to the room, just ask)

BE ORGANIZED



IF YOU KEEP SAYING "BEAR WITH ME FOR A MOMENT", PEOPLE TAKE A WHILE TO FIGURE OUT THAT YOU'RE JUST SHOWING THEM RANDOM SLIDES.

xkcd.com/365



BE ORGANIZED

- ❖ You have a story to tell; think about the narrative!
- ❖ The audience members are the passengers
- ❖ You are the driver!

- ❖ Think about what order of topics is most coherent

TIME MANAGEMENT



- ❖ Don't be done in 5 minutes
- ❖ Don't be 25 minutes in
and halfway done



START AT THE VERY BEGINNING

- ❖ Don't assume audience knows everything you've been doing
- ❖ Provide context and motivation
- ❖ Provide clear overview of project goals
- ❖ But...

...you want plenty of time to dig deeper.



START AT THE VERY BEGINNING

- ❖ Final presentation? Still start with project overview
- ❖ Second semester? Still start with project overview
- ❖ Don't assume that everyone saw (and remembers)
details from an earlier presentation



THEN DIG DEEPER

- ❖ Explain what you've accomplished
- ❖ Explain what you haven't (yet?) accomplished

- ❖ But... not time to explain EVERYTHING during presentation
(that's what deliverable is for!)

- ❖ Use your judgment about where to spend your time

CONSIDER VISUALS

**IF A PICTURE IS WORTH
A THOUSAND WORDS**



**WHY DO MEMES
HAVE CAPTIONS?**



HOW MUCH TEXT?

Do you really think that your audience will want to read this much text on the screen at once? And even if they are willing to read this much text, that surely means that they aren't actually listening to what you're saying at that moment. Sure it might feel safer to have everything you want written down on the slide so that you don't have to remember it, and perhaps it even gives you the opportunity to hide some Easter eggs for your own amusement (please touch your finger to your nose if you are reading this), but it just isn't as good of a live presentation if they are reading your slide and not listening to what you are saying.

HOW MUCH CODE?

```
def remove(self, posn):
    """Remove the item at the given iterator."""
    if not isinstance(posn, self.iterator):
        raise TypeError("Must provide valid iterator for remove")

    self._size -= 1
    walk = posn._nd
    if walk.left is None or walk.right is None:
        self._easyDelete(walk)
    else:
        # use predecessor as sub for the current node
        sub = walk.left.subtreeMax()

        # fix pointer from above
        if self._root is walk:
            self._root = sub
        elif walk is walk.parent.left:
            walk.parent.left = sub
        else:
            walk.parent.right = sub

        # relocate sub and remove walk
        if sub is not walk.left:
            # clean up below
            sub.parent.right = sub.left
            if sub.left is not None:
                sub.left.parent = sub.parent
            # sub takes over left child of walk
            sub.left = walk.left
            walk.left.parent = sub
        # sub takes over right child of walk
        sub.right = walk.right
        walk.right.parent = sub
        # sub gets new parent
        sub.parent = walk.parent
        # restore heap property from sub downward
        downward = True
```

```
while downward:
    child = sub.left
    if sub.right is not None and (child is None or sub.right.priority < child.priority):
        child = sub.right
    if child is not None and child.priority < sub.priority:
        self._rotateUp(child)
    else:
        downward = False

def _rotateUp(self, walk):
    """Rotate node walk up one level.

    Assumes that walk is not the root (but parent may be)

    """
    parent = walk.parent
    grand = parent.parent
    walk.parent = grand
    parent.parent = walk
    if parent.left is walk:
        parent.left = walk.right
        if walk.right is not None:
            walk.right.parent = parent
        walk.right = parent
    else:
        parent.right = walk.left
        if walk.left is not None:
            walk.left.parent = parent
        walk.left = parent
    if grand is None:
        self._root = walk
    else:
        if grand.left is parent:
            grand.left = walk
        else:
            grand.right = walk
```



PROJECT TIMELINE

- ❖ Make sure to explain what you've accomplished (and when)
- ❖ Make sure to explain what steps are still ahead of you
(and timeline for completing those tasks)



TEAM RESPONSIBILITIES

- ❖ Discuss individual roles
- ❖ Discuss how team functions/communicates/collaborates



DEMO

- ❖ Most groups save demo for the end
- ❖ But if early demo helps audience better understand, go for it

DEMO

- ❖ Technology will fail!
- ❖ Come prepared with screenshots/videos



ANY QUESTIONS?

