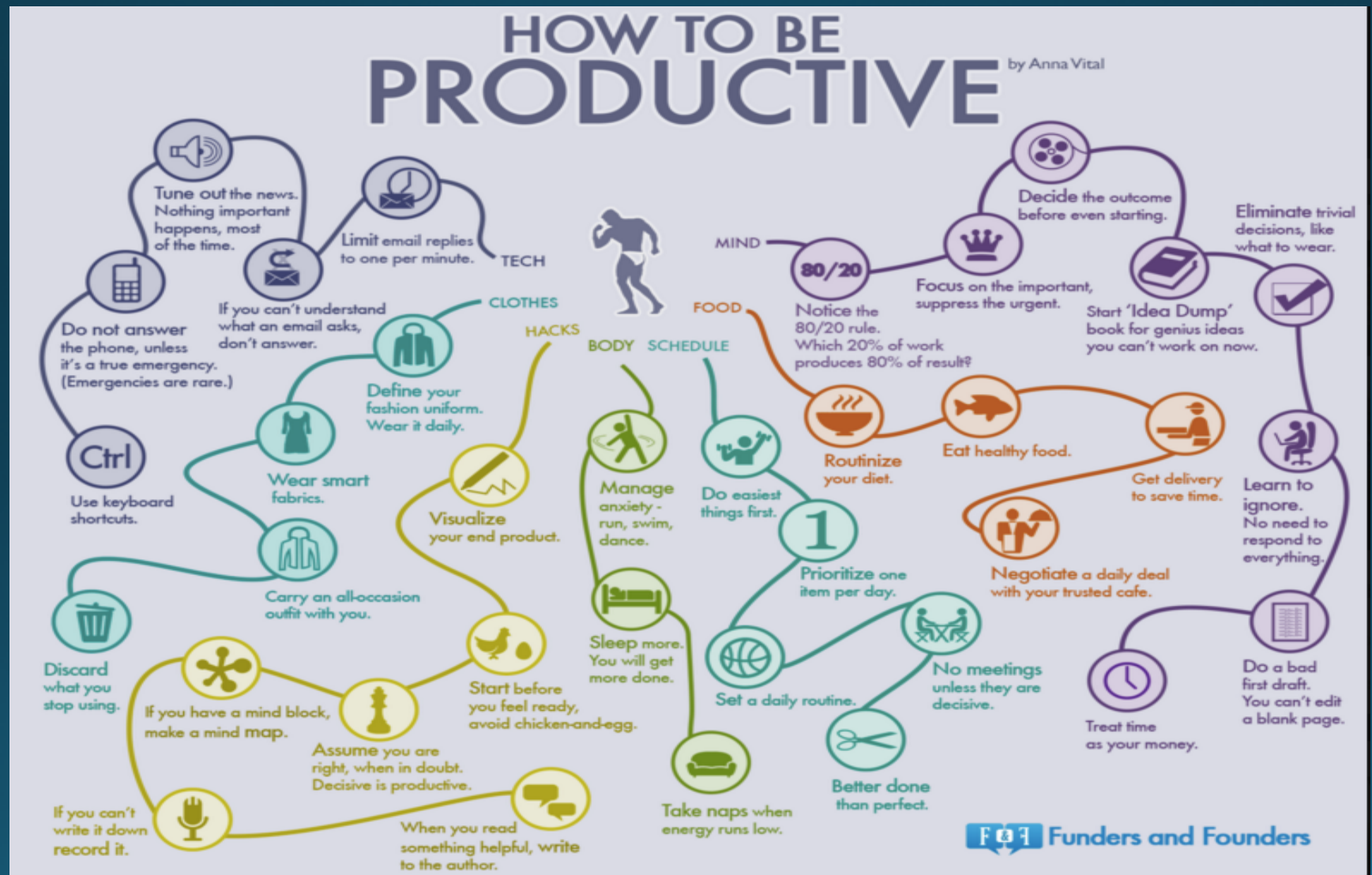


Today's Agenda

- Last Time in CSE 112
 - Participation Opportunities
- Brief Reviews
 - Blue Team
 - Gold Team
- Peter starts revealing some wants
- The Lecture
 - Group Efforts!

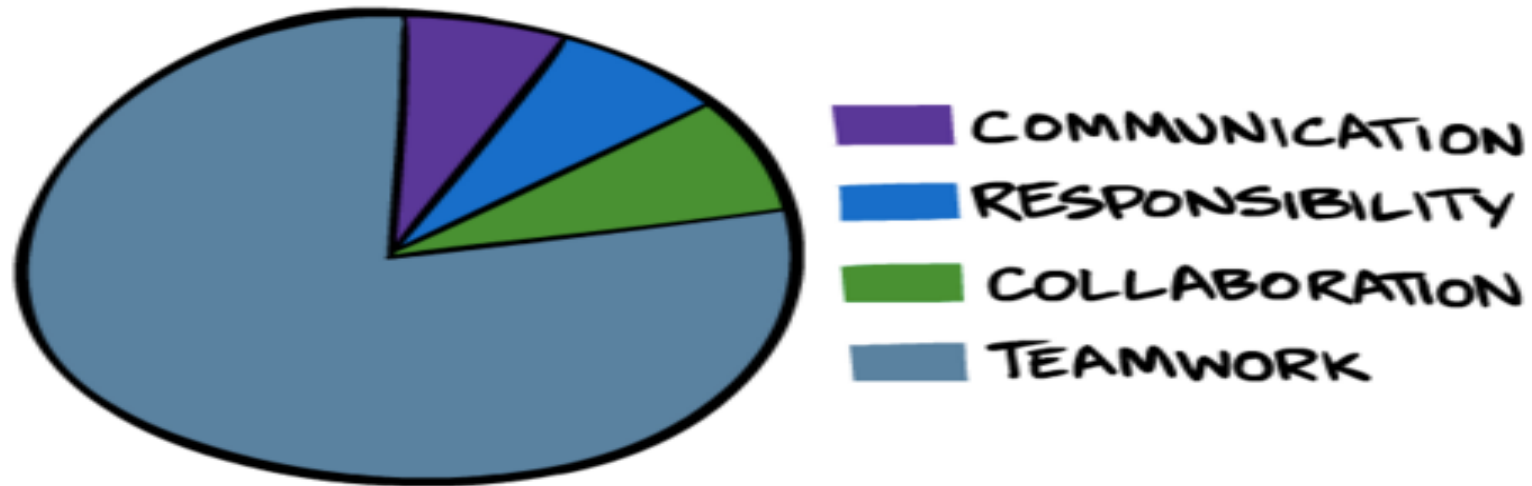
Let's Discuss



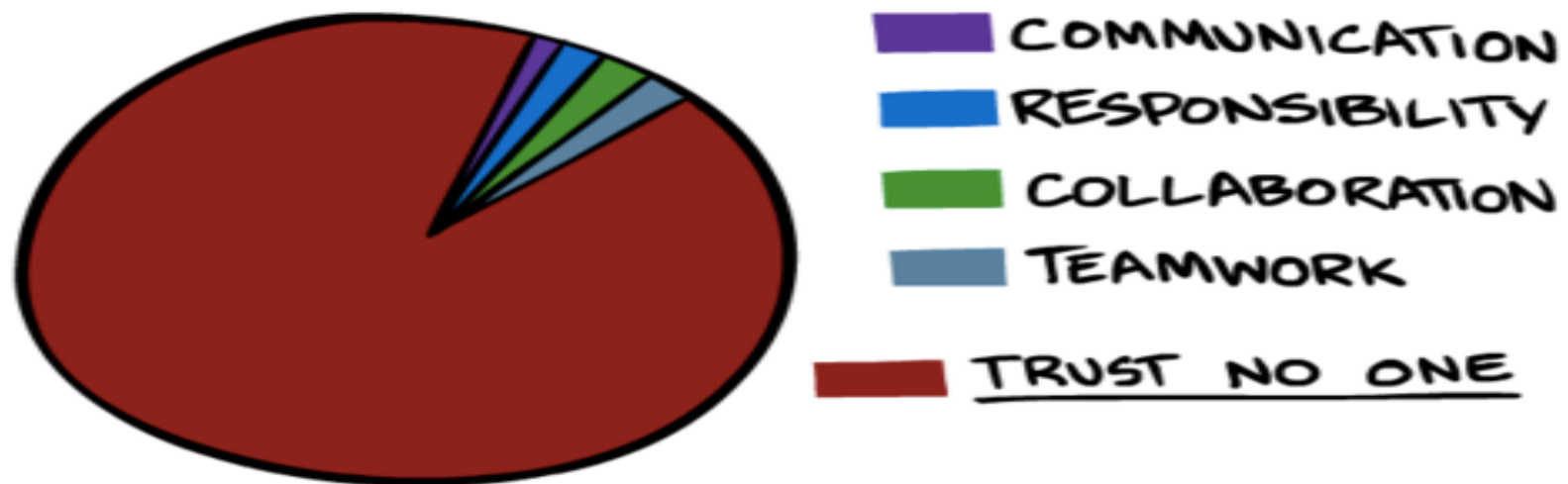
Lecture 3

On Groups

WHAT GROUP PROJECTS ARE SUPPOSED TO TEACH YOU



WHAT GROUP PROJECTS TAUGHT ME

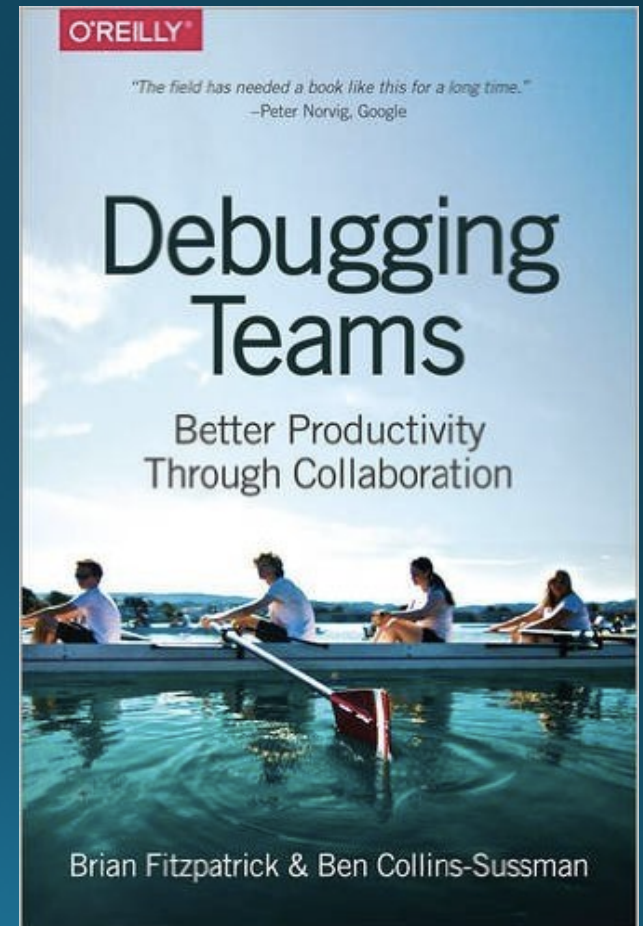


The Sad Reality of Group Projects

- Reality: Forced Collaboration
- Positive Point: Test your metal, learn from mistakes
- Negative Point: Unprepared for the experience to likely succeed
- Truthfully the Professors' have a hand in this too as we rarely give you any significant amount of coaching on how to actually succeed
- Let's take that on today

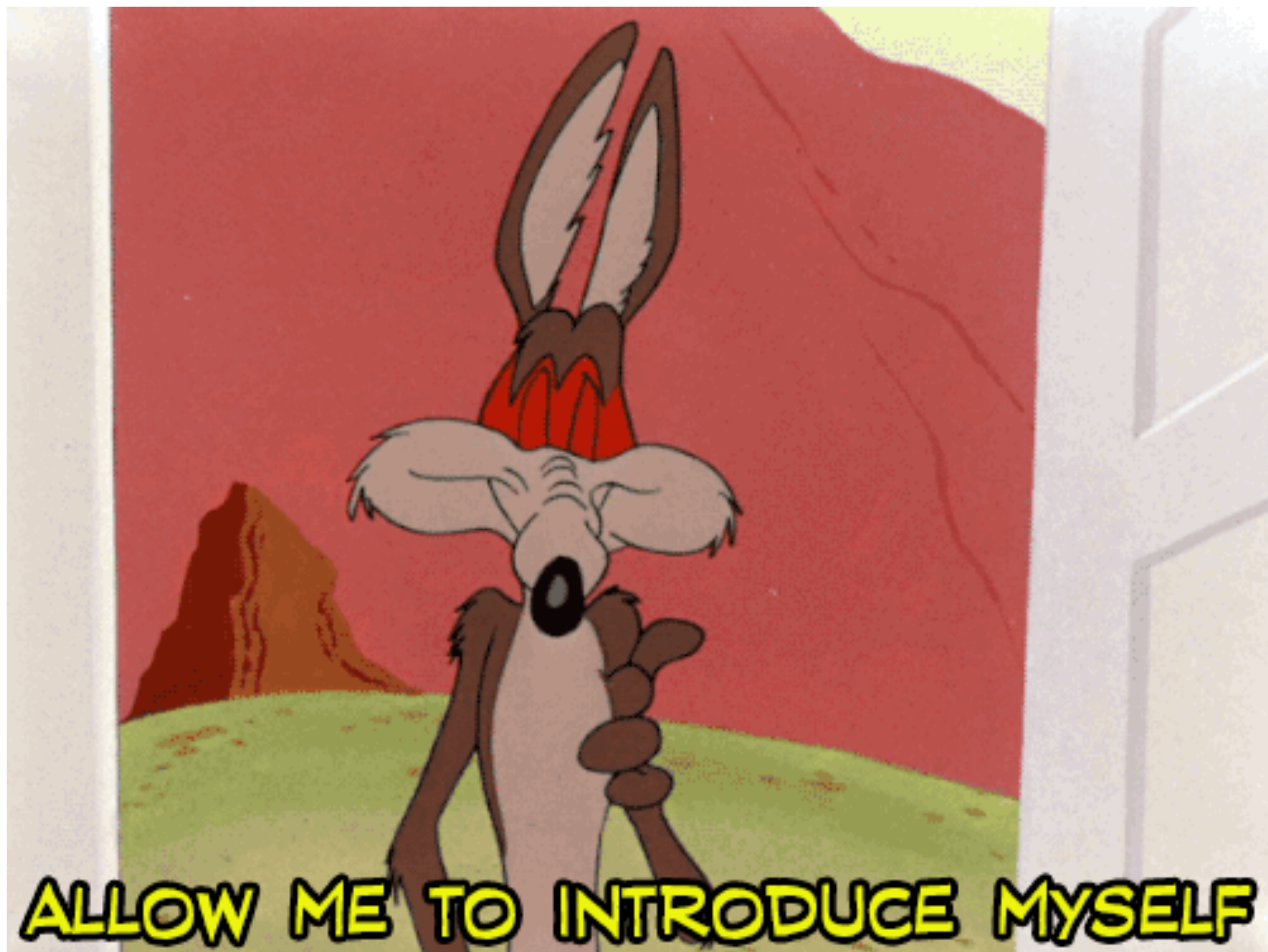
Debugging Teams

- If you believe some of the premises from last lecture the secret weapon a software engineer needs to master is often softer than code!
 - Much of the content out there focuses on programmer self-improvement. Following Postel's Law for yourself that makes sense!
 - Today's content tries to address team issues, though it starts of course with you again
- These slides represent some of the findings from *Debugging Teams* by Brian Fitzpatrick & Ben Collins-Sussman with numerous items from other books like *Joel on Software* and the Prof's experience mixed in for good measure



Geniuses

- A lot of our problem in groups starts with ourselves as we saw last time. Let's start with the rough root cause of many problems for programmers
- Subscribing to the Myth of the Genius Programmer
- Makes sense in a perverse way if we really equate our self-worth dominantly with what we code
- Unfortunately the emphasis on self is a myth as even recognized geniuses are nothing without a team
 - Jordan, Jobs, Gates, Torvalds, Musk, etc. without their team(s) they are nothing



Hiding

- When we subscribe to this apparently noble intent bad things can happen
 - From last time: leads to the impostor syndrome in extreme
 - Less extreme leads to code hiding and lack of communication
 - Ex: "I don't want people to see what is clearly a work in progress, otherwise they might judge me!"
 - Ex: "I am not sure my idea is any good so I'll just say nothing"
- Hiding also eventually leads us to information hoarding, which then builds us into fake 'geniuses' and may land us job security by information obscurity.
- Our intentions while self-protecting end up backfiring on us



Bus Factor

- Bus Factor – A numeric value that represents the number of people on the team/project that if hit by a bus would doom the project
- A low Bus Factor represents extreme project risk!
- Bus Factors often grow with info hiding and hoarding
- Sometimes teams accidentally worsen their bus factor
 - “Jennifer is the best at the build scripts, we will just let her work on it since that is most efficient”
 - “It doesn’t make sense for everybody to do everything, so we will each specialize ourselves”
- Bad Engineering at its’ heart – there is no redundancy or fail safe at the human level!

Bus Factor Story

Senior Engineer X: I really think I can do better than NodeJS, it will be your PHP in JS idea. Let's do this and build something amazing.

Repeat for 2 years with demos

Me: Ok, let's do it

Senior Engineer X: Awesome! About time, let's do this.

6 months pass, code is created with a bus factor near 1 and baked into systems and then...

Senior Engineer X: I am quitting. I'm bored and this isn't going anywhere.

- Result: Project was doomed and looked to never really work in a deep way without significant time/\$ investment. To recover 1 -2 developer year+ effort to unroll in-house code and revert to NodeJS

Programmer Life Cycle?

New Hires start at step 1

1. Ambitious
2. Productive
3. Irreplaceable → BUS FACTOR RISK!
4. Resentful → QUIT
5. Bored → QUIT
6. Unproductive → FIRED

Is this a microcosm of how some group projects go?

THIS PLACE WILL FAIL ONCE I'M GONE!



SAID THE GUY WHO NEVER DOCUMENTED ANYTHING

You Have to Work With Others

- Any sane organization shouldn't let you become the invaluable hoodied hermit developer
 - Some will but they shouldn't and things can and do go bad
- You can do it on your own clearly but you shouldn't
- You learn from others which makes you become a better programmer
- You can get more accomplished with a team than by yourself

Working alone is just riskier than working with others ultimately!

HRT (HEART) Method

- Humility – you aren't perfect, you should be open to self-improvement and growth
- Respect – you appreciate others, their intentions and their efforts. You treat them as fellow human beings with feelings just like you
- Trust – You trust that others are competent and will do the right thing if you let them

HRT Visualized



HRT doesn't solve, it explains

- Social conflicts generally can be traced back to a combination of these three pillars
- Understand that since there are many layers here it requires that everyone acknowledge these pillars otherwise it doesn't do much
 - Translation: You can employ this but others may not, if they don't they should be ejected from the team, collaborators group, organization, or user population

Tips on Living HRT

- Lose the ego
- Giving and receiving ~~criticism~~ feedback
- You aren't your code
- Fail fast and iterate
 - Not just for code but for social interactions
- Think Binary
 - Change the scale and make it a 0 and 1 outcome
 - Something simple, did you do it yes or no not something large that leaves room for failure
 - Doorknob turning ("yeah, it opened!") vs a self-critical discussion of doorknob turning ("my wrist wasn't at a 37degree angle and it took more than 1second and plus I wasn't really happy about it")

More Tips

- Leave time for learning
- Be patient
- Learn to listen
- Be open to influence
 - Changing your mind isn't weak ,it's mature
 - Say I don't know
- Think Soccer, Dempsey may have scored the goal but the Sounders won not just Dempsey
- The team plays not just you the "star" player

Meetings

- General Tips
 - Only invite people who **need** to be there
 - Have an agenda and share it before the meeting start
 - Be time aware, start and end the meeting on time, if done early, end early
 - Don't let the meeting wander (too much – allow outlets)
 - Back meetings against interrupts or unproductive time slots
- The stand-up meeting form
- In-person vs. Online
 - Non-verbal communication
 - Example: Remote only vs Local and then remote
- Maker's Schedule vs Manager's Schedule
 - Managers – interrupt for status
 - Makers - 4 hour blocks or more!
 - Headphones as your office door?

Programmer Communication

- Synchronous vs Asynchronous
 - Online allows for asynchronous, real world not so much, tends to be rude
- Meetings
 - In-Person
 - Online – GotoMeeting, Skype, etc.
- Group Chat
- 1-1 Chat
- Email
- Email List and Discussion Thread
 - Noisy minority problems
- Design Doc
 - A living document, not stone tablets
- Issues
- Code Comments
- Reviewing Comments

A fundamental mistake

- Your product / program is ultimately about communication, not just with a machine
 - Machine – yes it needs code that is syntactically correct, but in some sense it doesn't care how well developed your code is. It doesn't care, it's not alive. You and other humans care about runtime efficiency, optimization, maintainability, etc.
 - Developers – you and other people who read your code and work with you on your code
 - Stake Holders – people who guide the product and thus your code and likely fund your code
 - Collaborators (vendors, interested third parties, loose stakeholders) – people who have some vested interest in your code
 - End users -

WHAT ELEGANT CODE?

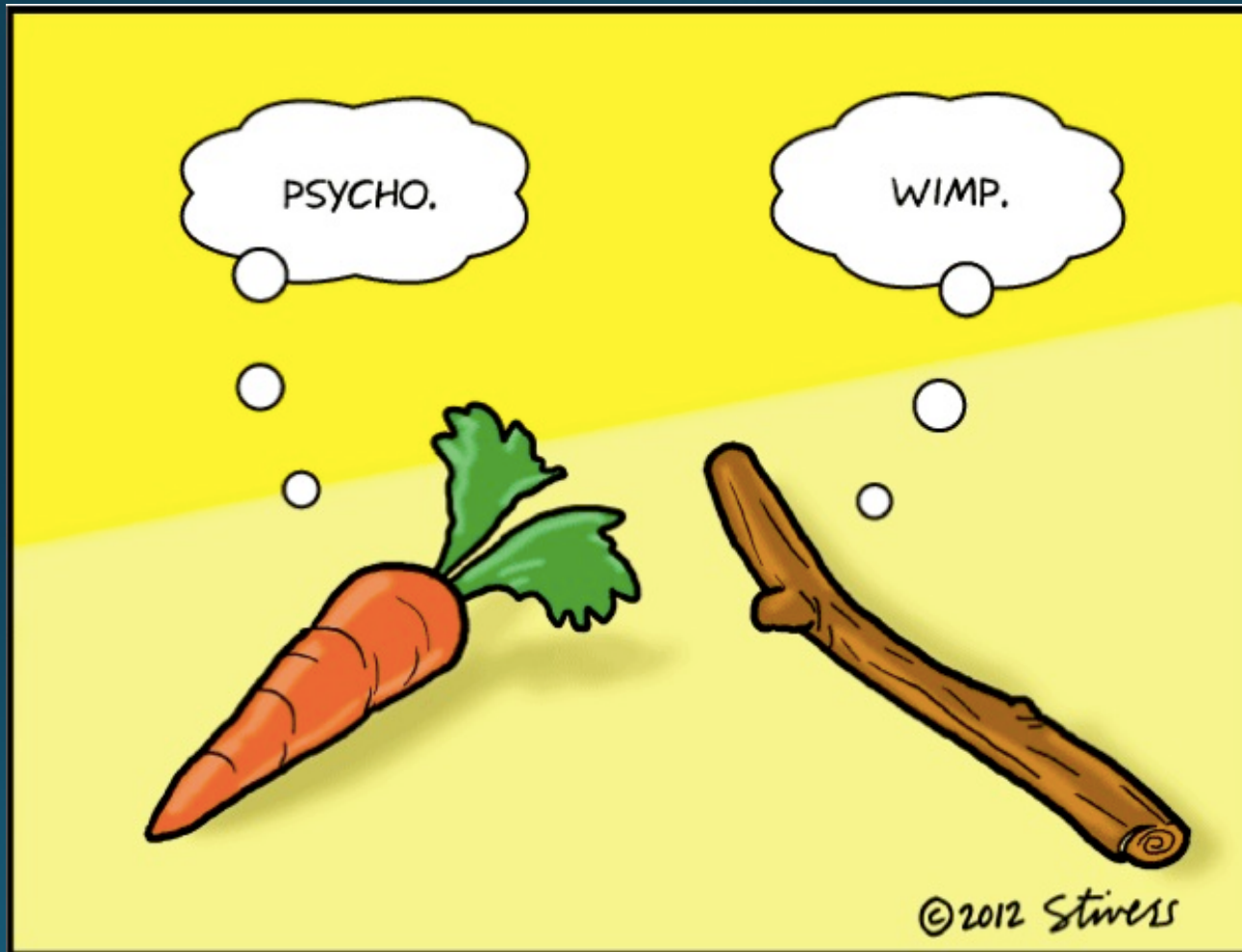


I JUST SEE A BUNCH OF ZERO AND ONES

Manager Leader

- We need them, without someone in charge we drift around pulled by random currents
 - Reality: A leader will generally happen either by appointment or self-selection
- Even if you never plan on being a manager/leader which is quite fine you would be wise to understand them
 - Helps you understand their decision making process
 - Leads to better communication and often better reviews (think raises)
- Good idea to understand motivation
 - <https://www.youtube.com/watch?v=u6XAPnuFjJc>
 - This might defy traditional motivation techniques (next slide)

Stick and Carrot



Being a Manager

- Being a leader is tough. It really isn't for everyone, but like all things it will take practice
 - Great opportunity to do so with safety here in school!
- Code is tangible, there is a result and so inherent makers will find it satisfying
- Management lacks quick results and is difficult to measure and can be less obviously satisfying
 - Relatization: Less obvious growth except over time, have to know the signs to look for
- Being a manager allows you to scale yourself in some sense, help bring your version out maybe only way to do it
 - You would like to get us to Mars too I bet, Elon might actually be able to do it ... strike that Elon might get his team to do it!
- What kind of manager are you going to be?

THE 8 TYPES OF MANAGERS

NOW THAT I'VE
MADE NOISE AND
DUMPED ON EVERYONE,
I'M OFF AGAIN



THE SEAGULL

I PREFER
TIMES
NEW
ROMAN



THE MICRO

CAN YOU MAKE
SLIDES FOR MY
AWARD SPEECH
ON YOUR PROJECT?



THE SHOW BOAT

USE ONLY
THESE TALKING
POINTS WITH
MY BOSS



THE SPIN DOCTOR



THE ABSENTEE

HOPE YOU
DIDN'T HAVE
WEEKEND
PLANS



THE PITCH FORK

BEFORE WE DECIDE
THE OFFSITE THEME,
LET'S REVIEW THE
GARTNER HYPE CYCLE



THE PONTIFICATOR

I MADE
CUPCAKES TO
GO THROUGH
YOUR REVIEW



THE PLEASER

Management Changes by Level



Tips on Being a Decent Manager

- First and foremost you need to adopt the right attitude to be a great leader
 - servant leadership vs management
- Second you can't lose sight of your managed craft
 - Leading from the front so to speak or bags on contractor, coding manager, etc.
- More tips on trying to be a reasonable leader
 - Lower your ego
 - Be calm
 - Change is constant anyway, don't freak out!
 - Ask lots of questions

More Tips

- Listen
 - My master debugging technique for some team members
- Have the right attitude
 - Can do, as opposed to can't be done
 - Yes, but vs Yes, and...
- Praise in public and criticize in private*
 - * Sometimes public 'floggings' have a use, but generally only to bring group back in line usually after the anti-pattern of not dealing with low performers or letting a toxic person in the group
- Be a catalyst
 - Set up the situation, accelerate a reaction
- Allow failure if it leads to growth (personal or technical)
- Be a teacher and a mentor

And even more tips

- Set clear goals
- Be honest with people (and yourself)
 - Doesn't mean tell everything though!
- Criticism Sandwiches
 - Positive, Negative, Positive
 - Serve with the right amount of bread!
 - Some people need lots of bread, others much less, but in general too much bread and the meat is likely lost or avoided
- Track Happiness
- Delegate but get your hands dirty
- Seek to replace yourself (or at least have a vacation back-up!)
- Know when it is time to make waves
- Fake it until you make it...really
 - You are an phony, we all are, just don't tell anyone 😊

Tips Vary

- “People are like plants” theory
 - Cactus, orchid, etc. they each have different care/feeding instructions
 - I like to think of managers more as coaches, you should try to bring the best out the players you have. Respect each and their differences and adopt your style to what works for them
- The best motivation is intrinsic motivation, though if you have a billion dollars to share I’ll take it 😊
 - Supposedly this motivation comes from giving people autonomy, mastery and purpose
 - Autonomy ability to be involved and make choices
 - Mastery ability to grow and master skills
 - Purpose make an impact
 - The purpose one gets really blown out these days with changing the world. That’s great but self-awareness needs to be practiced. A social sharing app for food pictures isn’t going to change the world it may change the world of how people discuss dinner. Be precise and refine your words

Manager Anti-Patterns

- Don't hire pushovers
 - Be careful with non-pushovers though they can be toxic people in disguise
- Don't ignore low performers
- Don't ignore human issues
- Can't be everyone's friend
- Don't treat your team like children (nor prisoners)

Poisonous People Habits

- Lack of respecting other people's time
 - Not reading the docs before asking when they knew they could, assuming person is available on their odd schedule, etc.
- Ego
 - Overriding consensus decisions repeatedly often using negative logic and corner cases,
- Entitlement
 - Watch out here, age and generational differences abound!
- Immaturity
 - I don't agree with this one but there is a point when SirHacksAlot's excessive giphy posts get on the team's collective nerves
- Paranoia
 - Yeah the "company" or "management" IS out to get you
- Perfectionism
 - See last lecture

IF YOU BITE IT AND YOU DIE, IT'S POISONOUS



VIA 9GAG.COM

IF IT BITES YOU AND YOU DIE, IT'S VENOMOUS

Anti-Venom

- Effective X% of the time where X is an integer somewhere above 0 and likely well below 100
- Don't feed the trolls
- Check your own emotions
 - Remember self-Postel you can only control yourself
- Find the grain of truth in the bile
- Kill them with niceness
- Know when to give up
- Play the long game

Venomous Groups

- The culture of a group/organization can be poisoned as well
- We should try to change the culture if we can if not we need to get out lest it gets us
- First problem, ideal situations don't really exist at least for long!
 - Ever heard of Hedonic Adaption?
 - Things often look better from the outside
- Truthfully things tend to vary
 - Toxicity inside a good organization, oasis of sanity in bad ones
 - Toxicity can change over time, what starts out toxic may get ok with familiarity. Good things of course can be perverted as well

Practical Thoughts on Our Groups

- Team Name
- Mascot / Personality
- Team Mission Statement
- Team “By-Laws”
 - No a-holes is not a reasonable by-law folks!
- T-Shirts, stickers and group focused swag might just be digital
- Getting to know the people beyond the code
- Developing the witty back and forth or not
- War Stories* - come later

Assignment

- Peter wants to meet the groups. This Thursday he wants each group to come forward, tell us who they are, maybe show off something about their personality.
- The time in front of class is short (~5mins) so you need to practice and not everyone needs to speak
- Slides (very few) or a web page might help make things go easy
- This is your first group grade so take it seriously

Coming Up in CSE 112

- Participation Homework Review
- Professor's In-Class Tour of WebStorm*
- Review of Projects from Gold and Blue Team's efforts from last year
- Discussion of possible approaches going forward
- Peter's pressure to get going
- Users, requirements tools, coding guidelines, and more