Software Engineering at Google

How to work well on teams

How to work well on teams

- Start with one variable you definitely have control on; **yourself**
- For a creative collaboration, we need to reorganize our behaviours around the core principles of humility, respect and trust
- Spend less energy with people-related problems and more time writing quality code

Help me hide my code

"Can you make it possible to create open source projects that start out hidden to the world and then are revealed when they’re ready?"

Reason? Insecurity

People are afraid of others judging their work in progress

The Genius Myth

The Fantasy:

- You are struck by an awesome new concept
- You vanish into your cave
- You then “unleash” your software on the world
- Your peers are astonished by your genius
- People line up to use your software.
- Fame and fortune follow naturally.

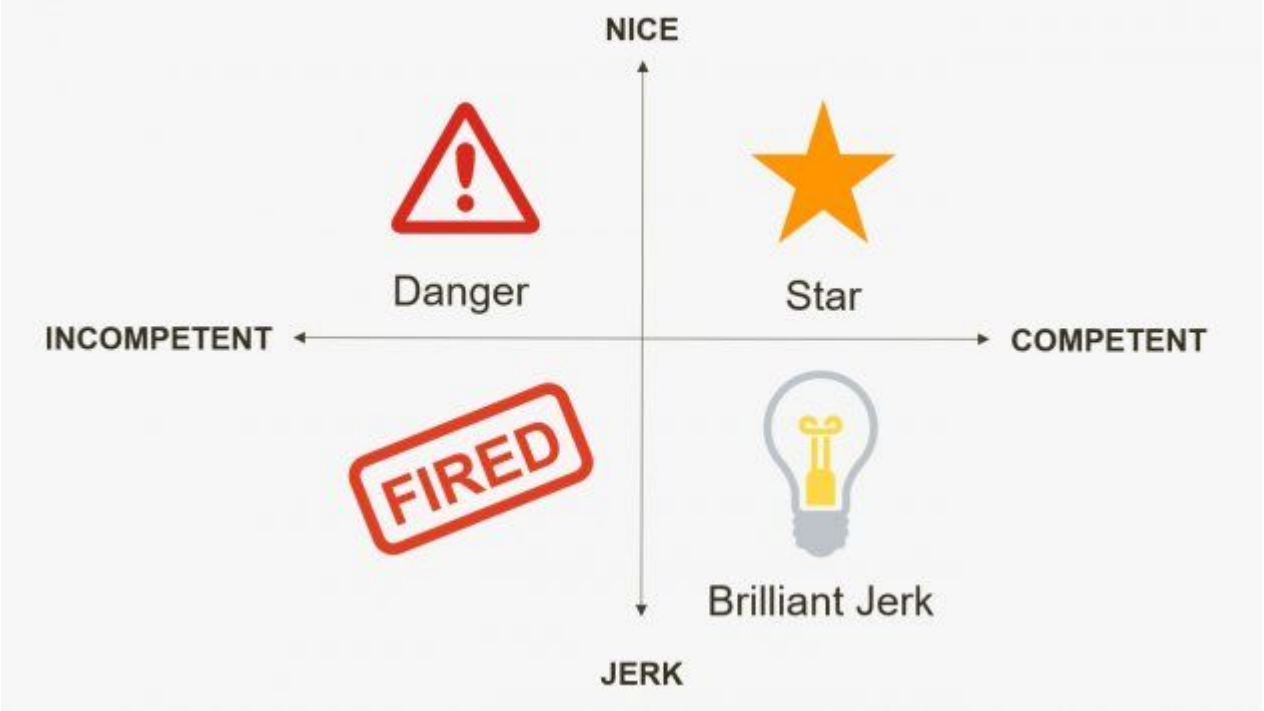
The Genius Myth

The Fantasy:

- You are struck by an awesome new concept
- You vanish into your cave
- You then “unleash” your software on the world
- Your peers are astonished by your genius
- People line up to use your software.
- Fame and fortune follow naturally.

But hold on: time for a reality check. You’re probably not a genius.

The Genius Myth



Hiding Considered Harmful

- Early Detection

  Gamble, Risk of reinventing the wheel

- The Bus Factor
- Pace of progress

  Big softwares are made in great teams

Pillars of Social Interaction

Humility

Respect

Trust

Humility, Respect, Trust: **Lose the ego**

Nobody wants to work with someone who behaves like they’re the most important person even if they are.

Being humble vs being a doormat

Go for collective ego instead

Humility, Respect, Trust: **Learn to give/take criticism**

Code review culture

Criticize the work not the person

- You’re not your code

Care about the end goal

Humility, Respect, Trust: **Learn to give/take criticism**

“Man, you totally got the control flow wrong on that method there. You should be using the standard xyzzy code pattern like everyone else.”

Antipatterns:

- Accusing them of being wrong
- Demanding change for no reason
- Make them feel stupid

Humility, Respect, Trust: **Learn to give/take criticism**

"Man, you totally got the control flow wrong on that method there. You should be using the standard xyzzy code pattern like everyone else."

Antipatterns:

Accusing them of being wrong

Demanding change for no reason

Make them feel stupid

"Hey, I'm confused by the control flow in this section here. I wonder if the xyzzy code pattern might make this clearer and easier to maintain?"

Humility, Respect, Trust: **Fail fast and iterate**

Accept your mistakes and build experience

> “I just spend $10 million training you”

Failure is an option

GoogleX

Humility, Respect, Trust: **More**

Blameless post-mortem culture

Learn patience

- Value conflict

Be open to influence

- Don’t be stubborn
- Admit mistakes and show vulnerability
- Saying “I don’t know” isn’t that hard

Being Googley

Googleyness:

- Thrives in ambiguity
- Values feedback
- Challenges status quo
- Puts the user first
- Cares about the team
- Does the right thing