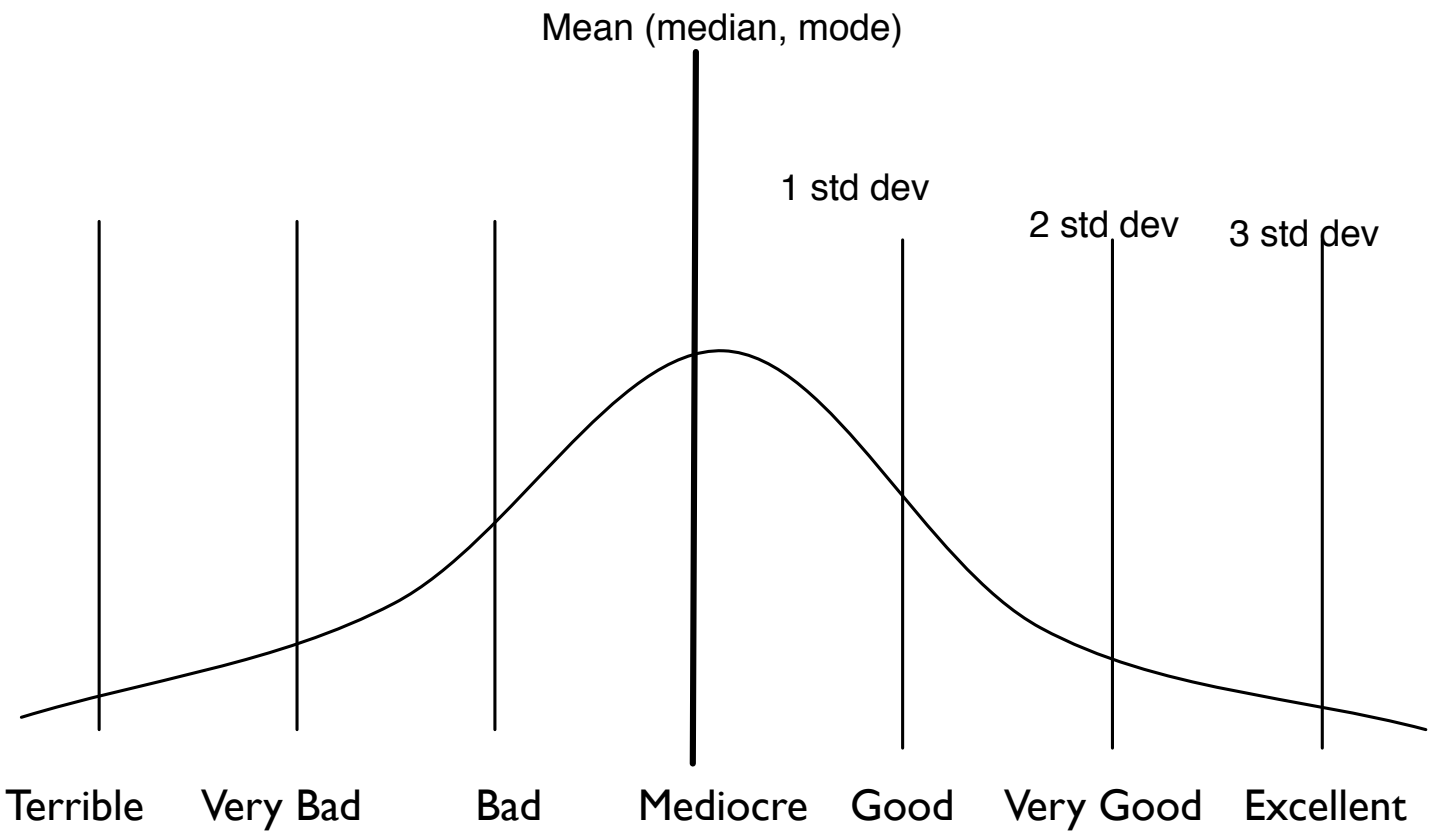
You Can Teach Problem Solving and You Should

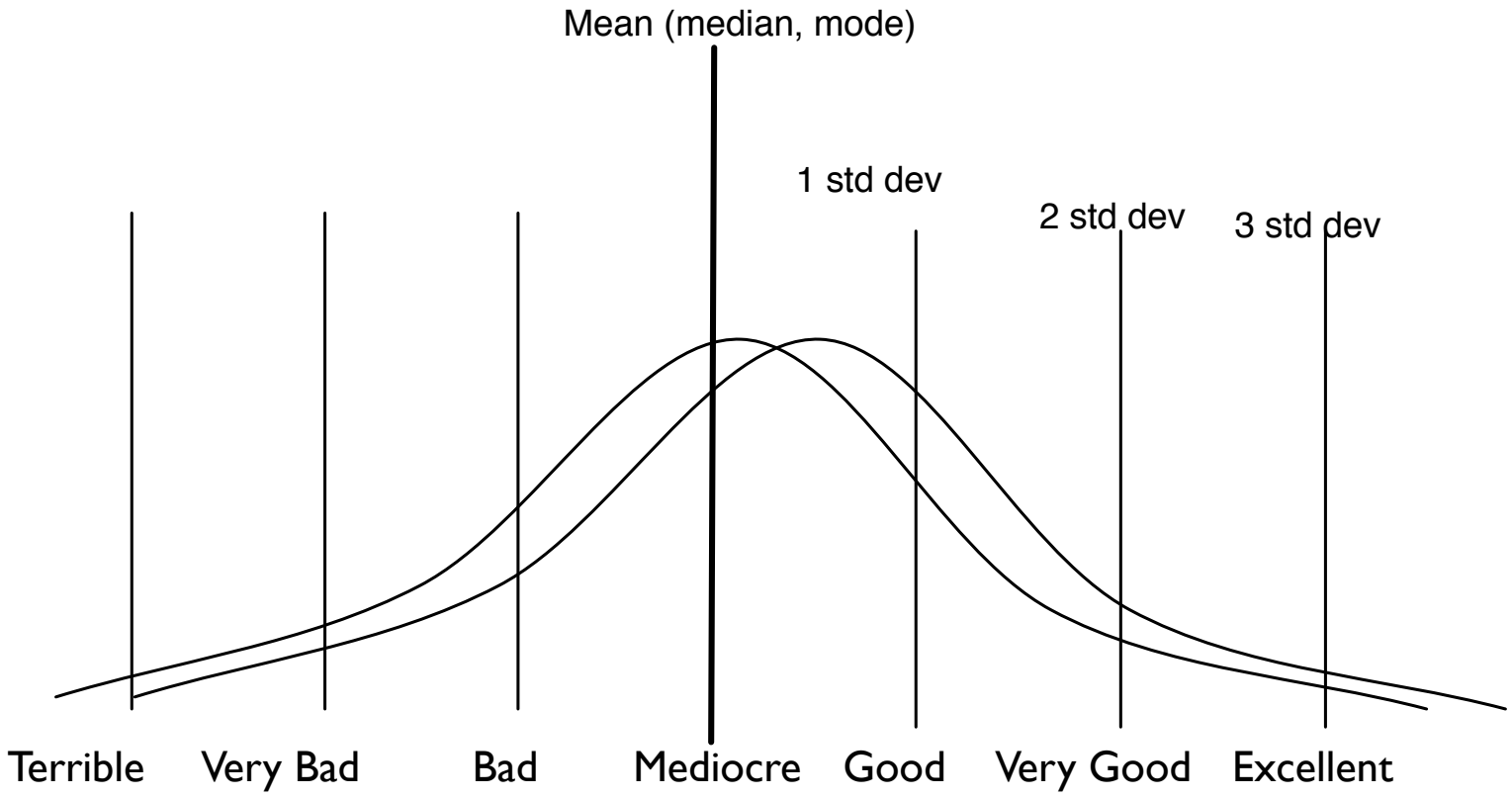
Elizabeth Zwicky
Great Circle, Inc

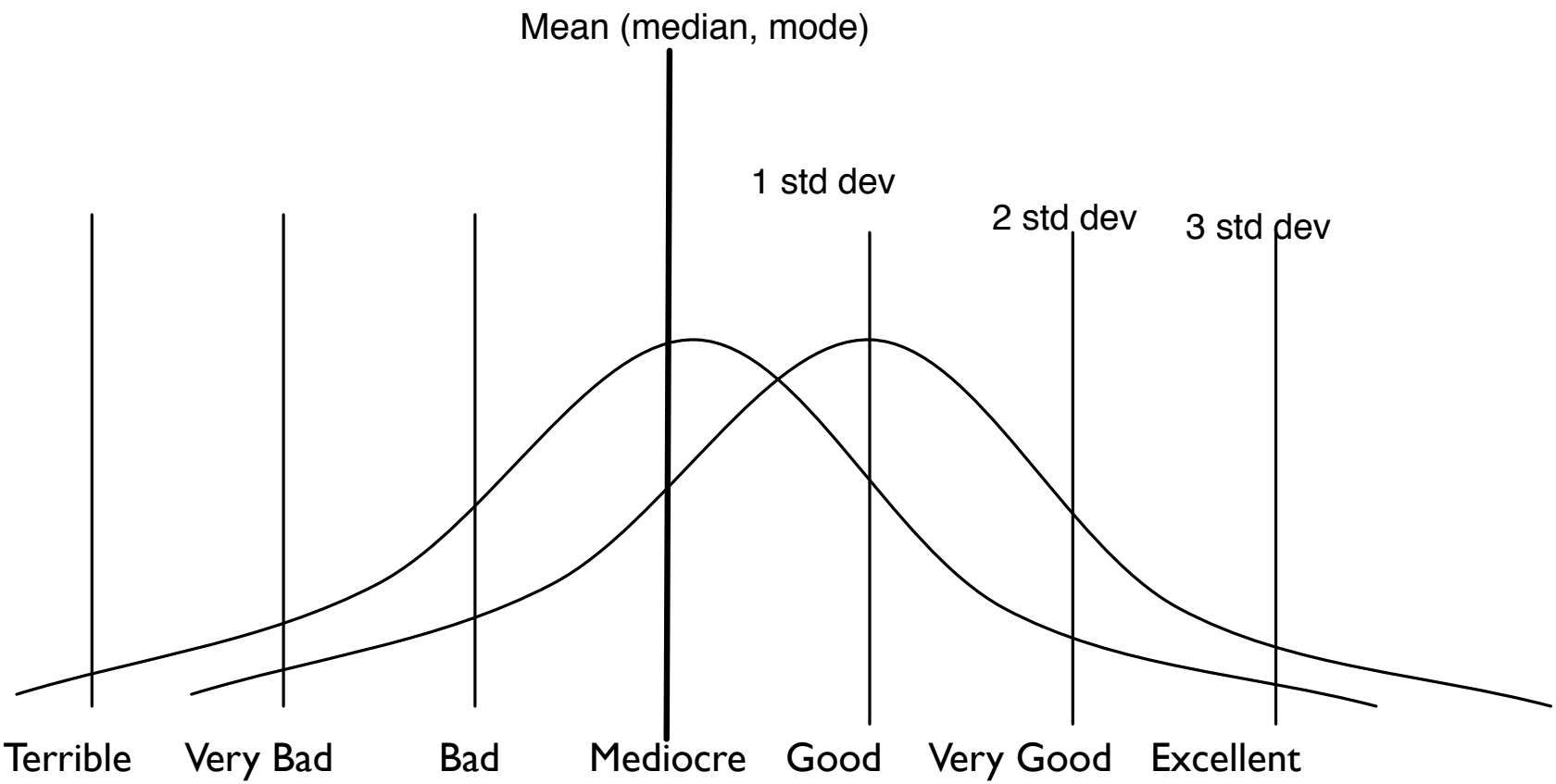
- Why do I think you can teach problem solving?
- Why do I think it's important that you believe you can teach problem solving?
- Why do I think it's important to teach problem solving?

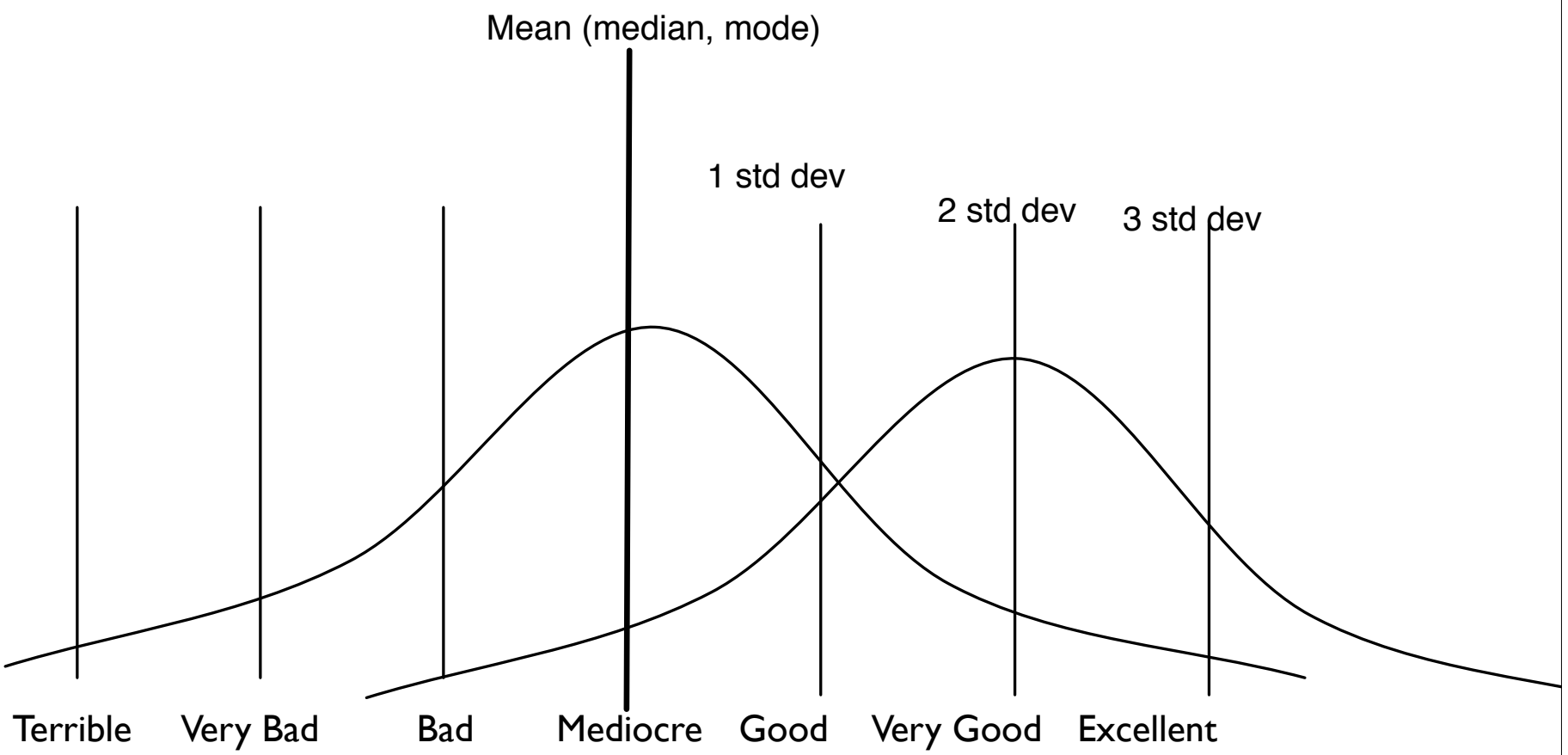
Why do I think you can teach it?

- Been there, done that.
- Actually, it's not particularly controversial.
  - What education wonks think of as problem solving is a slightly different skill set, but everybody in education believes that all the relevant skills are teachable.


Mean (median, mode)
1 std dev
2 std dev
3 std dev
Terrible
Very Bad
Bad
Mediocre
Good
Very Good
Excellent


Mean (median, mode)
1 std dev
2 std dev
3 std dev
Terrible
Very Bad
Bad
Mediocre
Good
Very Good
Excellent


Mean (median, mode)
1 std dev
2 std dev
3 std dev
Terrible
Very Bad
Bad
Mediocre
Good
Very Good
Excellent


Mean (median, mode)
1 std dev
2 std dev
3 std dev
Terrible
Very Bad
Bad
Mediocre
Good
Very Good
Excellent

Why is it important to think so?

- Stereotype threat, or, if you think you can't do it, you're not as good at it.
- Malleable intelligence and belief, or, not only can you change how smart you are, step one is believing it's possible.

Why do we think it isn't possible?

- Not much training – most system administrators are still natural talents who didn't need much teaching.
- School systems believe it can be taught, but mostly don't teach it.

Why is it important to teach problem-solving?

- Working with bozos is not fun. People who cannot problem solve will behave like bozos.
- Being able to solve problems improves people's lives.

Who can teach problem solving?

- Anybody can do it to some extent, but great tutors are rare.
- Good practitioners ≠good tutors.
  - Unconscious competence.
  - Teaching is a skill of its own.

When and Where?

- Can you teach old dogs new tricks?
  - Yes, but they have to want to learn them.
- Can you teach in a work environment?
  - Yes, but it's slow – the best teaching is resource-intensive.

How?

- Believe that it's possible, and communicate that belief.
- Teach problem-solving methods explicitly.
- Do effective tutoring.

Problem solving methods

- General problem-solving – approaches that apply to all problems.
- Specific techniques – ways of thinking about particular domains.

General approaches

- 7 +/- 2 steps
- Not perfectly general; different domains prefer different flavors
- All of them will include a stage where you figure out what the problem is and one where you verify that you solved it.

My favorite approach

- Identify the problem
- Analyse the problem
- Find solutions
- Choose a solution
- Implement the solution
- Verify the solution

Identifying the problem

- The complaint is what you're told.
- The symptom is what they're complaining about.
- The defect is what's actually broken.
- The problem is what you need to get working.

The Internet is Broken

- That's a complaint.
- The symptom is cnn.com: host not found.
- The defect is cnn.com is down.
- The problem is that the user needs news.
- OK, really the problem is that the user is bored.

Analyse the problem

- What are the rules of engagement?
- What do you know about the process when it works?
- This is the picture-drawing and searching question phase.

Find solutions & choose one

- Always aim to identify multiple solutions.
- Weigh the choices against each other.
- Consider side-effects and long-term effects.

Verify the solution

- Did the problem go away?
- Was it your fix that caused it to go away? (or, how to be smarter than a chicken)
- Is it going to stay gone?
- What would you do differently next time?

Specific techniques

- Every domain has key concepts and techniques:
  - Read log files.
  - Networks work like a stack – figure out what layer you're at.
- The concepts are important but the details aren't – 5, 7, 9 stack layers, who cares?

Being a good tutor

- Scaffolding and spotting.
- Conceptual focus.
- Praise and support.
- Verbalization.

Scaffolding and spotting

- Scaffolding is doing the absolute minimum to allow somebody to reach a higher level than they can reach alone.
  - Ideally, they don't really notice the help.
  - Questions are usually more useful than answers.
- Spotting is being unobtrusive but catching errors that would be too painful.

Conceptual focus

- Knowing that 2+3 = 3+2 is more important that knowing they both equal 5.
- Conceptual errors can be hard to spot, particularly if you can't control the problems.
  - Look for repeated errors.
  - Ask the student to explain concepts.

3 kinds of conceptual errors

- Wrong model
  - I want to be warmer fast so I'll turn the thermostat up.
- Bad problem solving
  - But the light is under the lamppost.
- No model

Praise and support

- Learning is inherently rewarding.
- Praise, but don't overpraise.
- Reassurance is often more important for a struggling learner than praise.
  - This is hard; it's not just you.
  - You are making progress.

Verbalization

- Putting things into words.
- Restating for the student what just happened.
- Getting the student to restate what happened.

Practice

- People will happily practice given:
  - A safe environment.
  - Problems at the right level of difficulty.
  - A continuous stream of problems.
- In a teaching context, providing these is mildly tricky. In a work context, it's very hard and involves faking it a lot.

Safe environments

- An environment is safe when:
  - Mistakes aren't punished.
    - Laughing at somebody is punishment.
  - Nothing is permanent.
- Virtual machines and dedicated training machines are usually safe.

Semi-safe environments

- If you can't dedicate resources to training, you can make a semi-safe environment.
- Mistakes still aren't punished, but are accepted as part of the learning process.
- Learners get low-stakes machines.
- Lots and lots of scaffolding and spotting.

References

- How To Solve It; A New Aspect of Mathematical Method, 2nd Edition, G. Polya, Princeton Science Library, 1988, ISBN 0 -691-02356-5
- The Logic of Failure; Recognizing and Avoiding Error in Complex Situations. Dietrich Dörner, Perseus Books, 1996, ISBN 0-201-47948-6

References 2

- Brain Power; Learning to Improve Your Thinking Skills, Karl Albrecht, Simon and Schuster, 1987, ISBN 0-671-76198-6
- Archimedes' Bathtub; the Art and Logic of Breakthrough Thinking, David Perkins, W. W. Norton and Co., 2000, ISBN 0-393 -04795-4

References About Design Problems

- de Bono's Thinking Course, Edward de Bono, Facts on File Publications, 1985, ISBN 0-8160-1895-2
- The Art of Problem Solving, Russell L. Ackoff, 1978m ISBN 0-471-85808-0

References: Puzzles to Play With

- aha! Insight, Martin Gardner, Scientific American, 1978, ISBN 0-7167-1017-X
- 100 Games Of Logic, Pierre Berloquin, Barnes and Noble, 1995, ISBN 0-7607 -1396-0