How to grow junior engineers in a world with LLMs

Daniel Wagner-Hall, **CodeYourFuture**
daniel@codeyourfuture.io

With LLMs, software engineering is a series of dialogues...

… But it always was!

Dialogues with precise vocabulary

- Dialogue with a product manager
- Dialogue with a user
- Dialogue with other engineers on your team
- Dialogue with other teams
- Dialogue with the compiler
- …
- Dialogue with an LLM

"With LLMs I can get stuff done so fast."

Experienced engineer

"With LLMs I can get stuff done so fast."

Junior engineer

"With LLMs I can get stuff done so fast."

But when something goes wrong I get stuck."

Junior engineer

"Junior engineers think they're going fast.

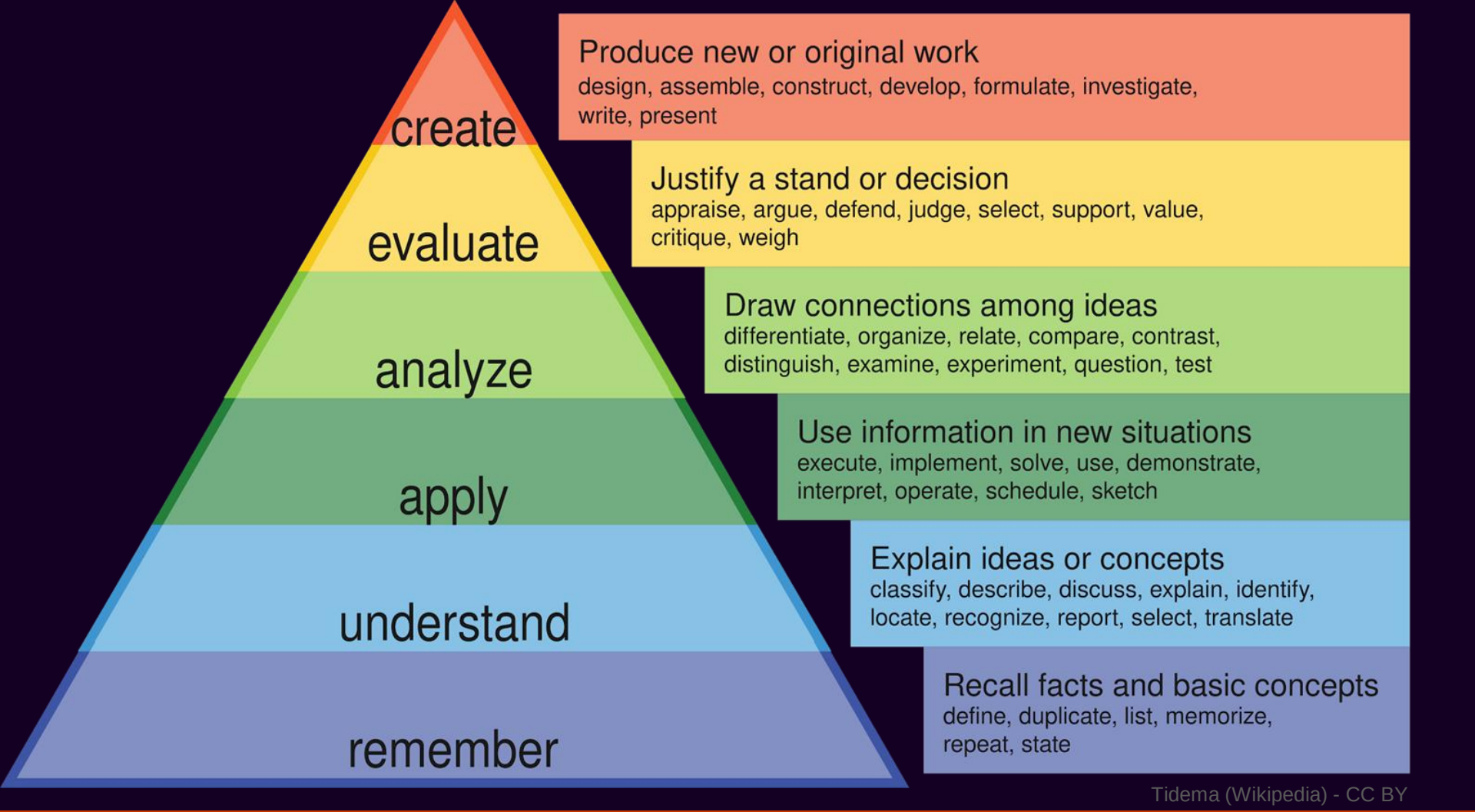
But they create bugs they don't even understand."

Experienced engineer

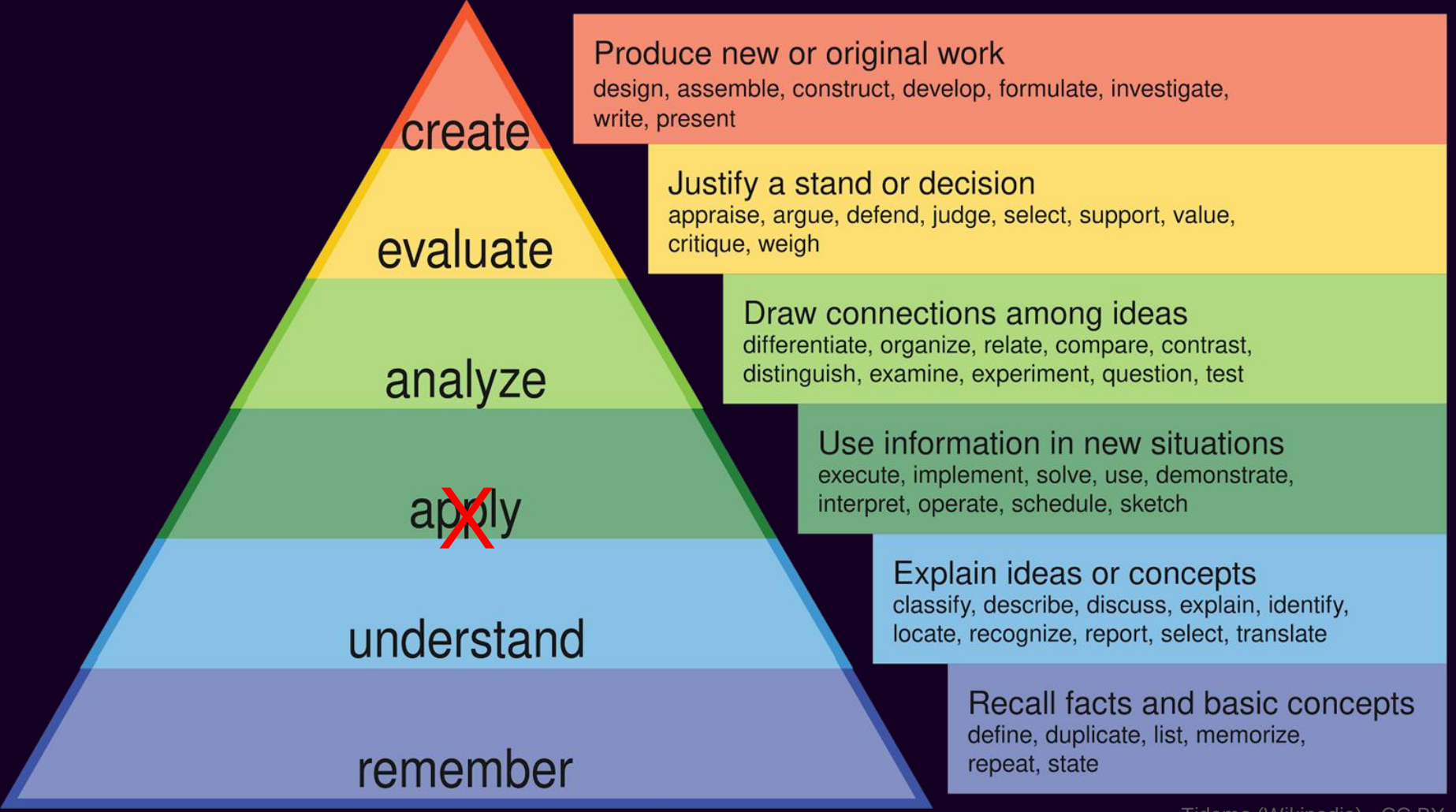
"I have built up good judgement over many years and know what can go wrong.

So I have the vocabulary to prompt effectively and review effectively."

Experienced engineer



Tidema (Wikipedia) - CC BY



Tidema (Wikipedia) - CC BY

Example Programming Vocabulary

- Global state
- Concurrent data modifications
- Pass by reference vs value
- Floating point
- Invalid pointers

Programming Skills

- Not actually obsolete
- Skipping applying with LLMs makes it hard to analyze/evaluate/design
- Possible approaches
  - Ban LLMs until you can analyze
  - Apply in a different way - e.g. "code review" as the core applying skill
    - But can't offload that to LLMs either!
    - And can't rush/skip it!

CodeYourFuture's Approach

A bit of both!

- Discourage LLM use when learning fundamentals
  - Self-guidance: "Will I need to ask the same thing again"
  - Objective assessments of understanding: Interviews
- Increasing project-based learning with mentors
  - Reading and talking about code as a core skill

How to grow junior engineers in a world with LLMs

Daniel Wagner-Hall, **CodeYourFuture**
daniel@codeyourfuture.io

