Things Software Developers Should Learn about Learning

Lauren Margulieux
Associate Professor
Department of Learning Sciences
This Talk is For...

• Students and Junior Engineers
• Engineers learning new tools or skills
• Mentors
• Recruitment and hiring
• Developers of new tools
Experts Recognize, Beginners Reason

The value of expertise is pattern recognition

**Takeaway:** If beginners can’t do something now, practice identifying patterns and practice the component steps to automate them.

*It's not that I'm so smart, it's just that I stay with problems longer.*

-Albert Einstein-
Predictors of programming ability are unclear

- Mindset: aptitude vs. practice
- Failed predictors of aptitude
  - Prior experience
  - Personal Characteristics
  - Preferences
- Best predictors related to rate of learning
- **Takeaway**: Many perceived predictors of programming ability have no empirical evidence
Understanding a Concept Goes from Abstract to Concrete and Back

**Takeaway:** Cycle between abstract concepts and concrete examples to learn.
The Internet and AI Have Not Made Learning Obsolete

The Internet and generative AI (OpenAI Codex, GitHub Copilot) are good at low-level details, but using them has a cost.

**Takeaway:** Developers still need to learn the conceptual information that is offloaded to tools.
What You Can Do to Support Learning

• Learners - Practice identifying and applying patterns
• Hirers – Consider how you predict programming aptitude
• Mentors - Provide concrete examples and abstract concepts
• Mentees – Aggregate examples to discover underlying principles
• Learn low-level skills before relying on AI tools
Paper coming soon to Communications of the ACM

lmargulieux@gsu.edu