

Swim! Don't Sink...

Why Training Matters to an SRE / DevOps Practice



Jennifer Petoff (aka Dr. J)
Twitter: @jennski
Global Head of SRE EDU at Google

Hello
my name is

Jennifer Petoff
(aka Dr. J)
Google Ireland

- Ph.D. in Chemistry
- 13 years at Google
- Lead the SRE EDU team
- Co-editor of the SRE Book
- Part-time Travel Blogger at Sidewalk Safari



Poll: What types of training experiences have you encountered?



**Why is training
important?**

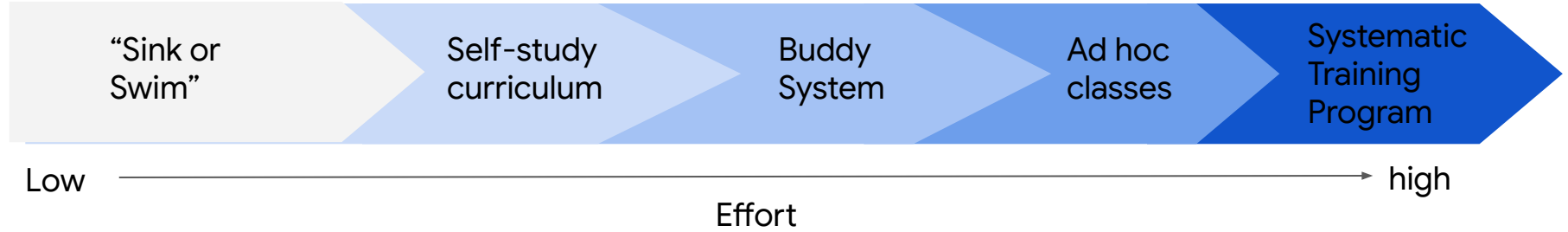


believe in
yourself

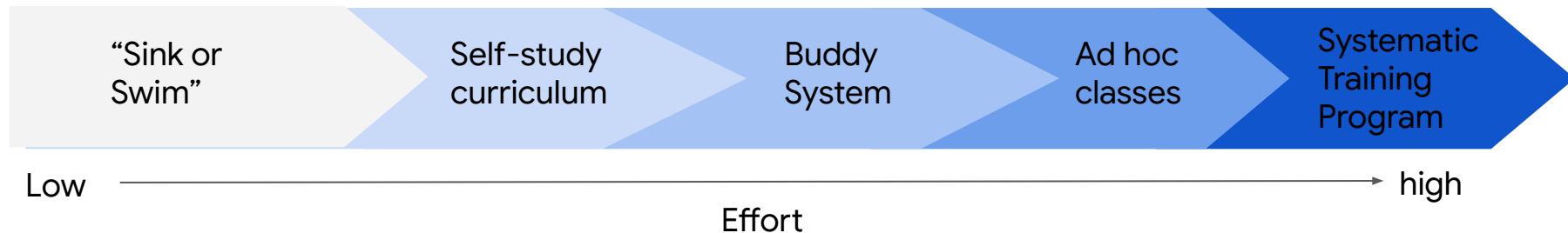




Continuum of Training Options

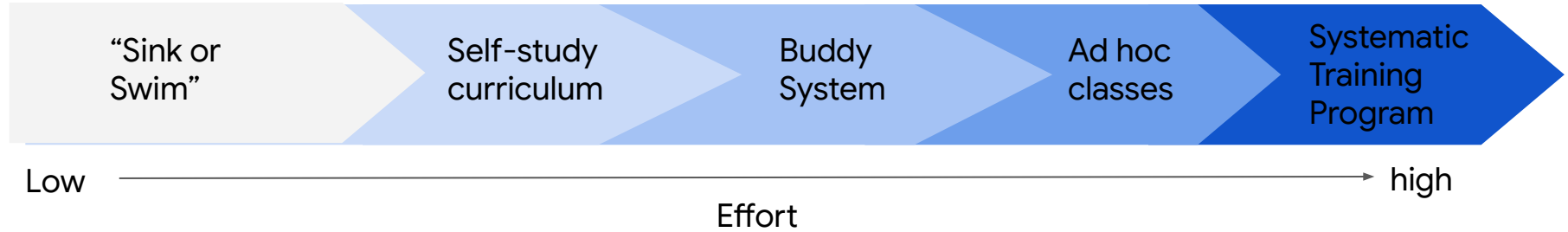


Continuum of Training Options



- Avoid “Sink or Swim” if you value inclusivity
 - Breeds stress, frustration, attrition
 - Imposter syndrome

Continuum of Training Options

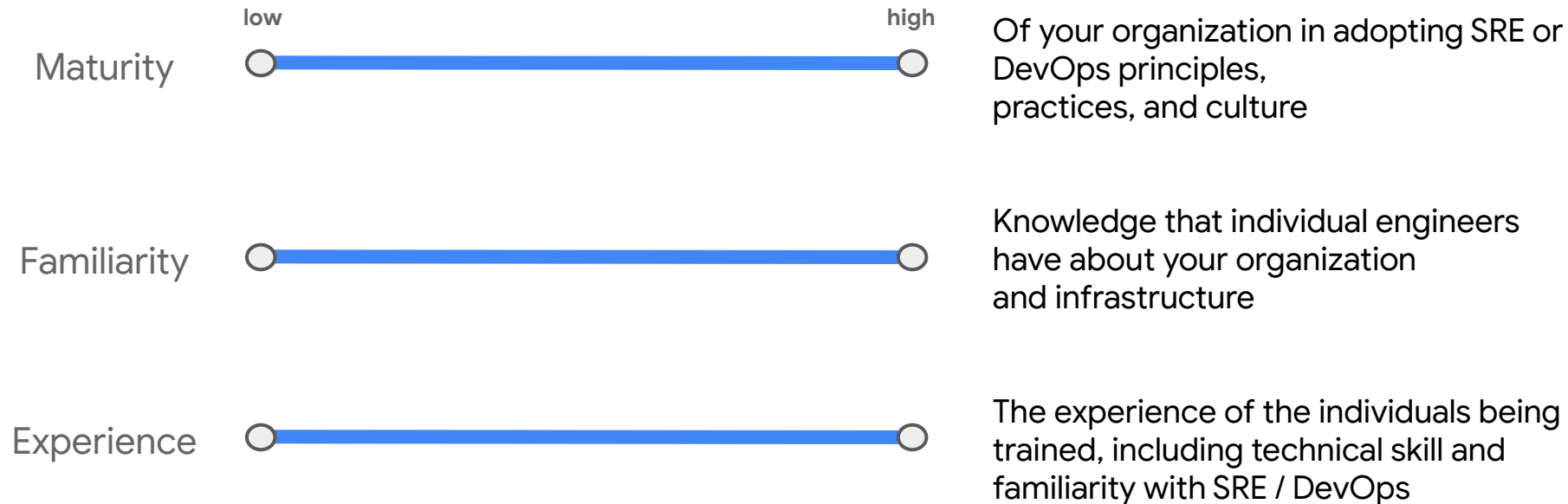


- Higher touch options signal:
 - Leadership commitment to the development of employees
 - Ensure everyone is speaking with one voice
 - Help imbibe the desired culture
 - Reinforce desired behaviors



**What should I
teach?**

Organizational Dimensions That Inform Training Needs

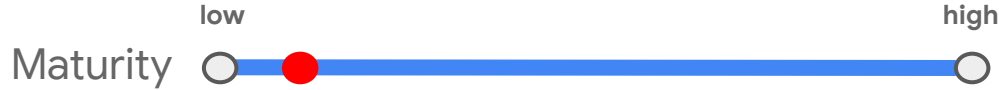


Poll: Who's in the Audience?



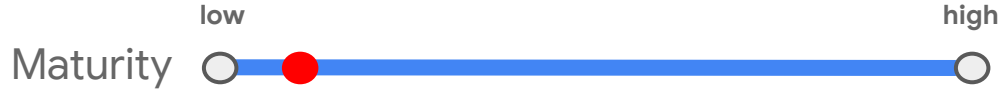


Step 1: Address Any Skill Gaps



- Does your team use a defect tracking system (e.g., Jira, Bugzilla)?
- Does your team have a project planning model (e.g., Agile)?
- Is work done in a version control system (e.g., Git)?
- Are solutions clearly articulated and vetted by team members (e.g., with design docs) before starting implementation?
- Does your team have a shared repository of commonly used tools and libraries?

Step 1: Address Any Skill Gaps

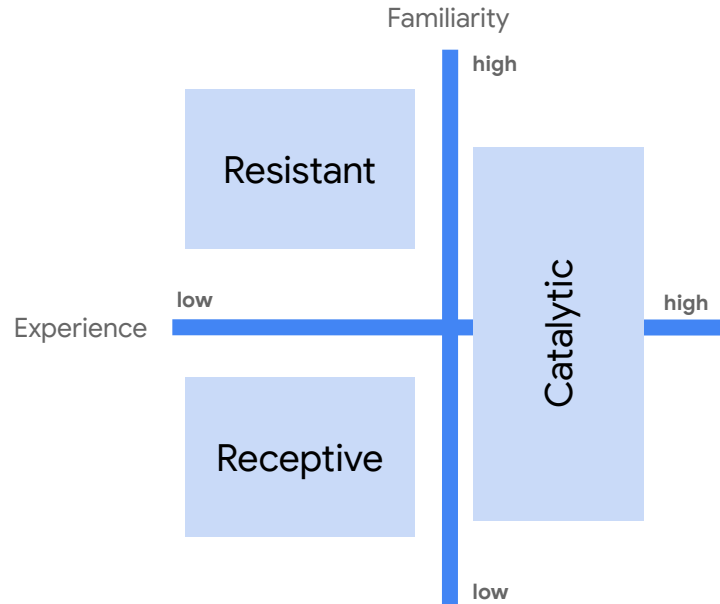


- Does your team use a defect tracking system (e.g., Jira, bugs)?
- Does your team have a project planning model (e.g., Agile)?
- Is work done in a version control system (e.g., Git)?
- Are solutions clearly articulated and vetted by team members (e.g., with design docs) before starting implementation?
- Does your team have a shared repository of commonly used tools and libraries?

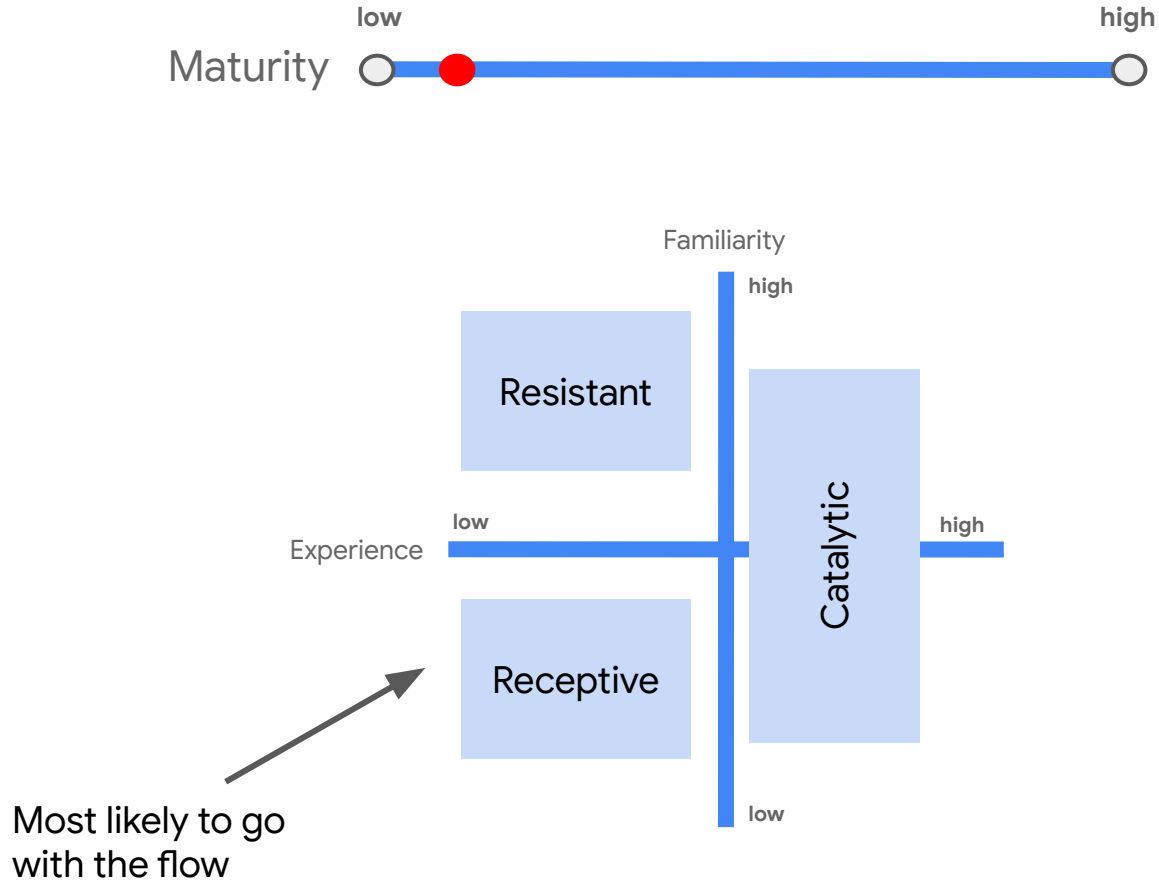
Start Here!



Step 2: Know Your Team & Tailor the Message



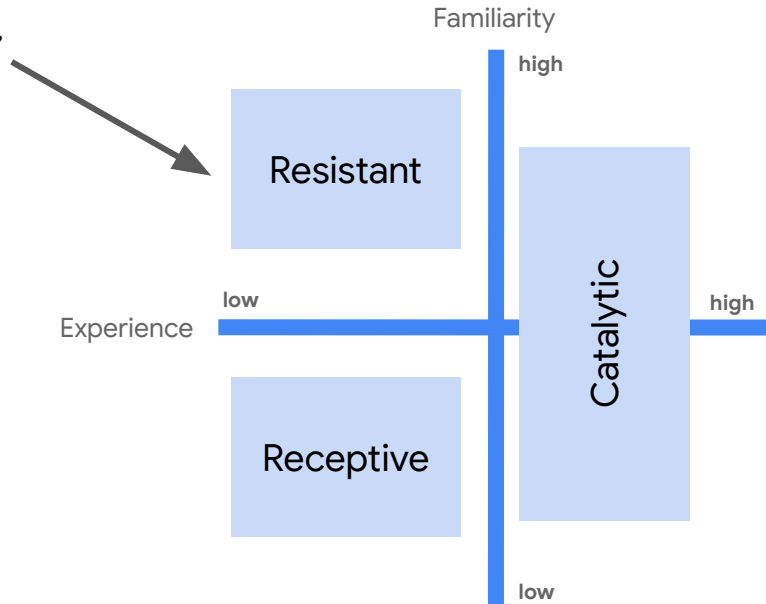
Step 2: Know Your Team & Tailor the Message



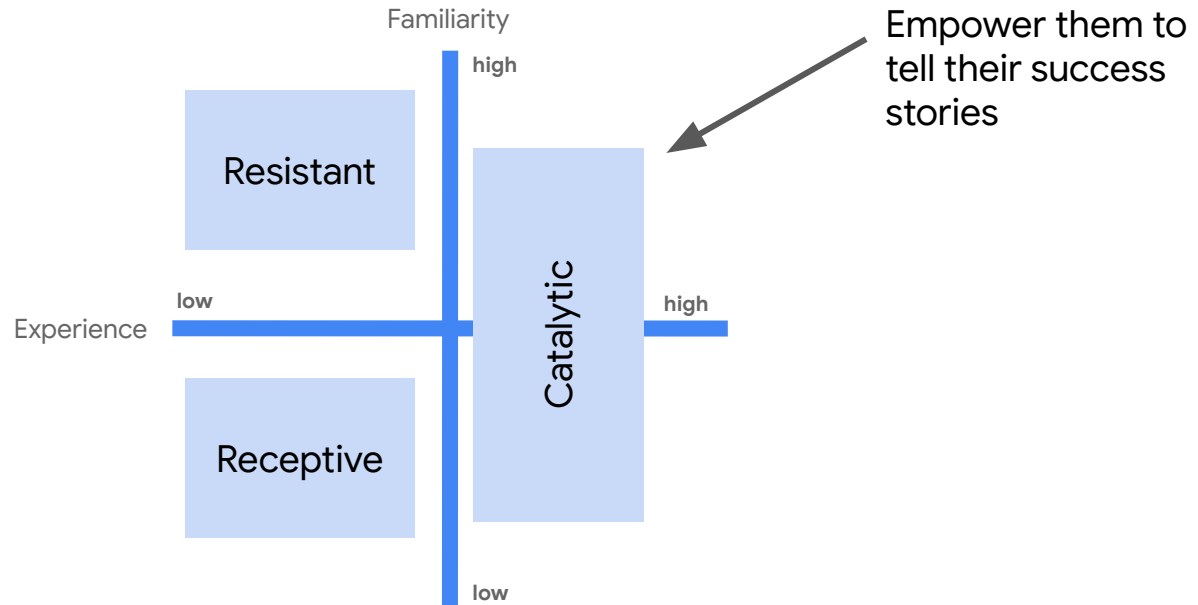
Step 2: Know Your Team & Tailor the Message



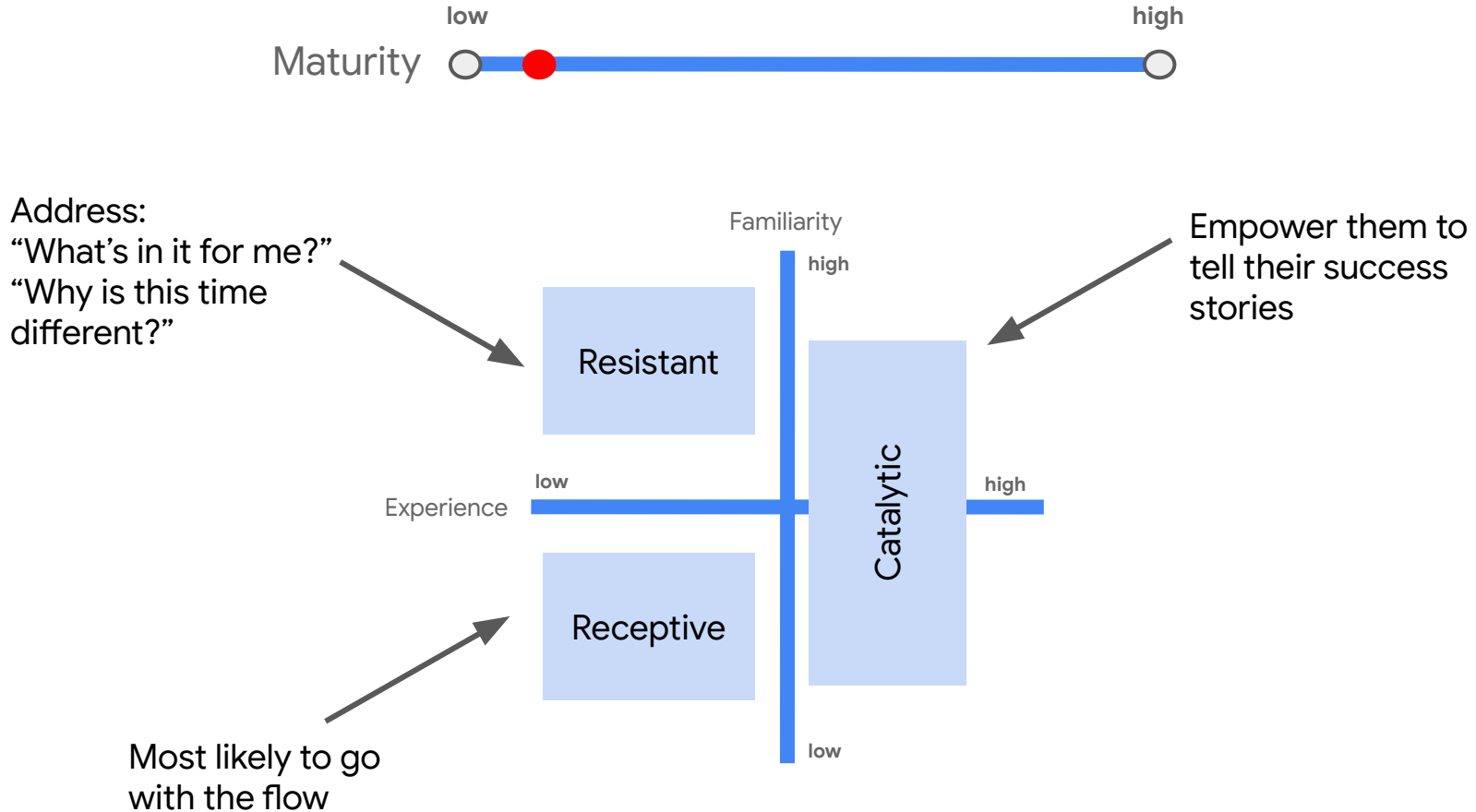
Address:
“What’s in it for me?”
“Why is this time
different?”



Step 2: Know Your Team & Tailor the Message

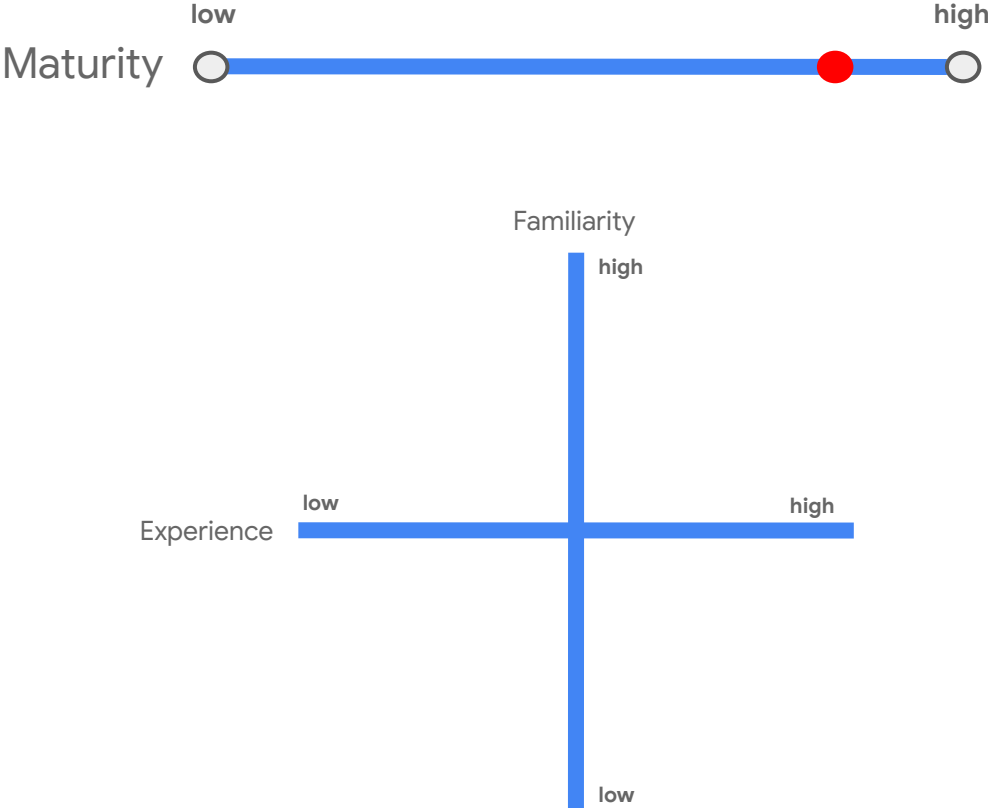


Step 2: Know Your Team & Tailor the Message

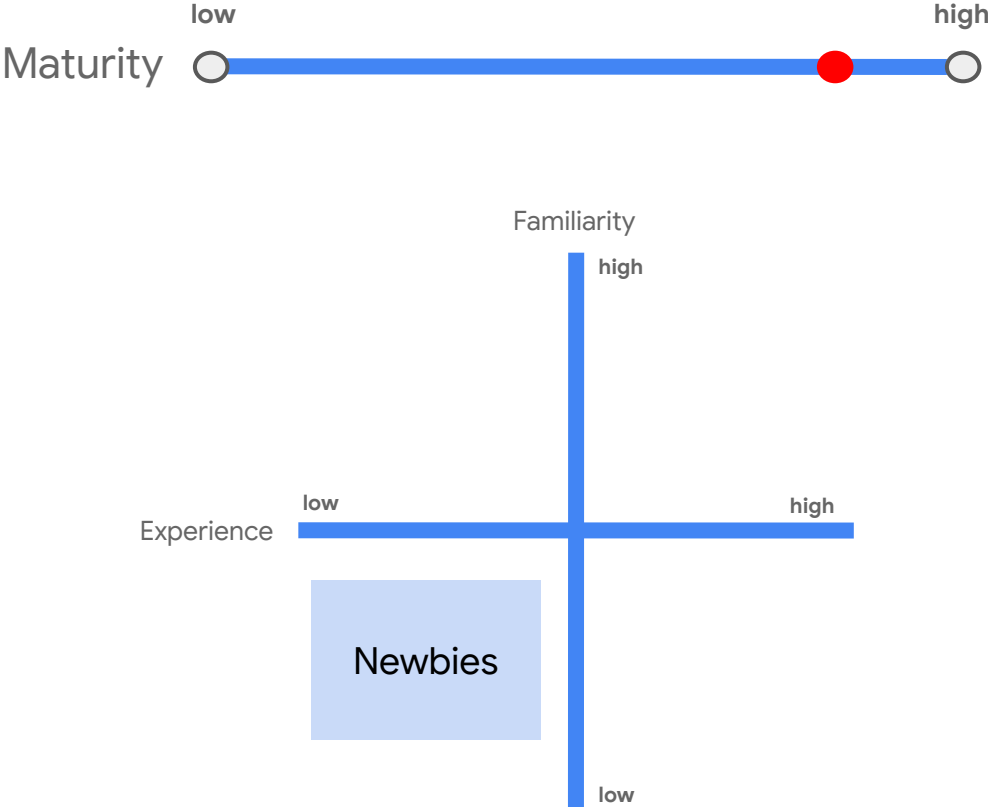




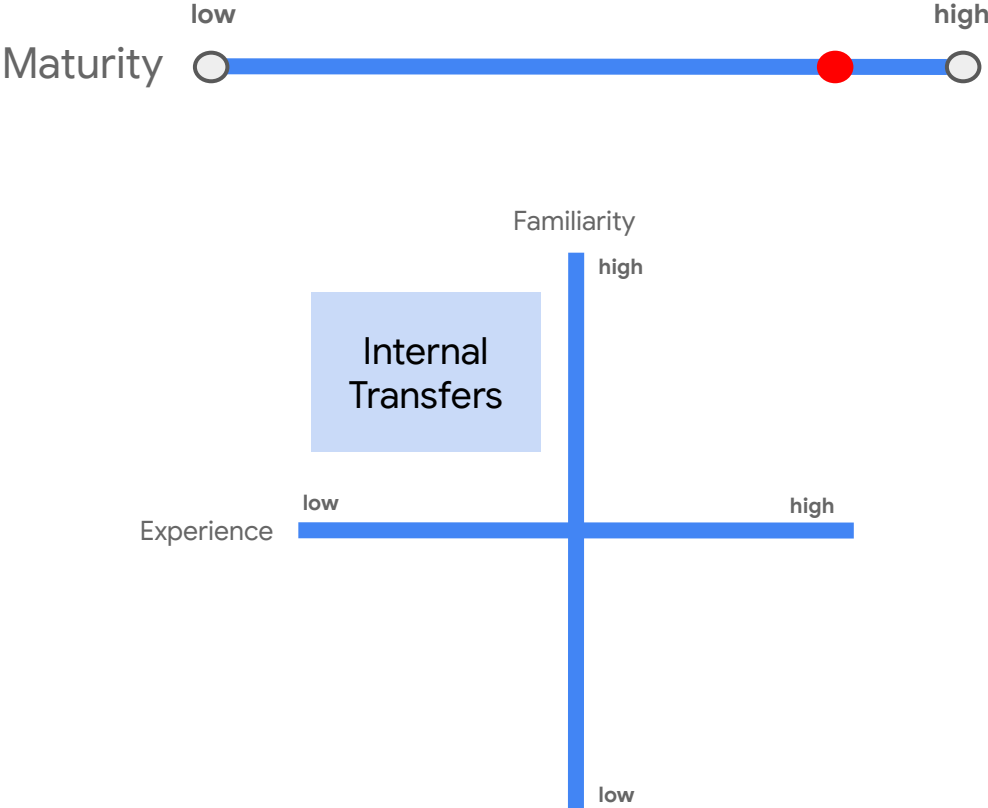
Step 1: Assess the Team Mix



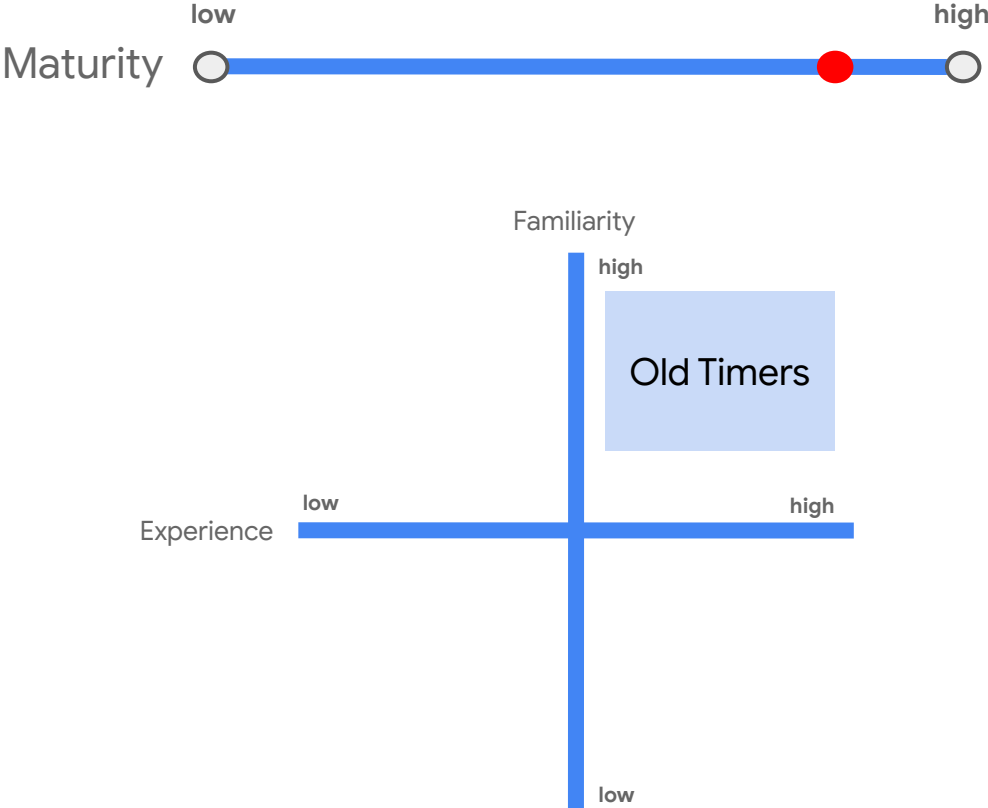
Step 1: Assess the Team Mix



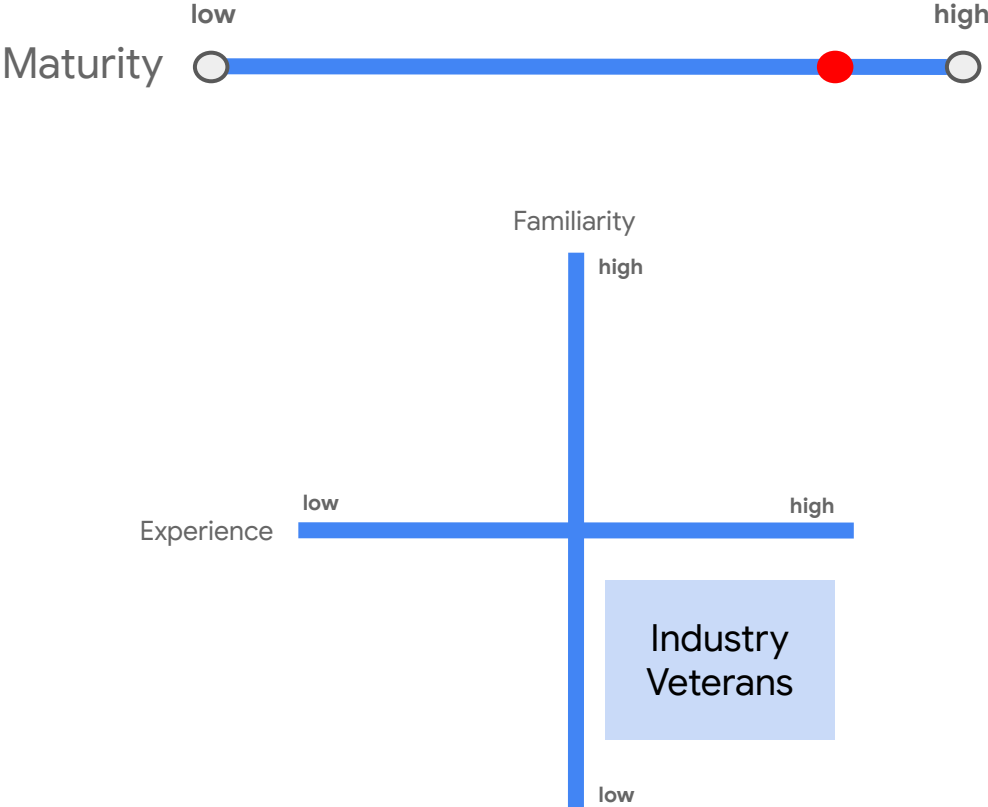
Step 1: Assess the Team Mix



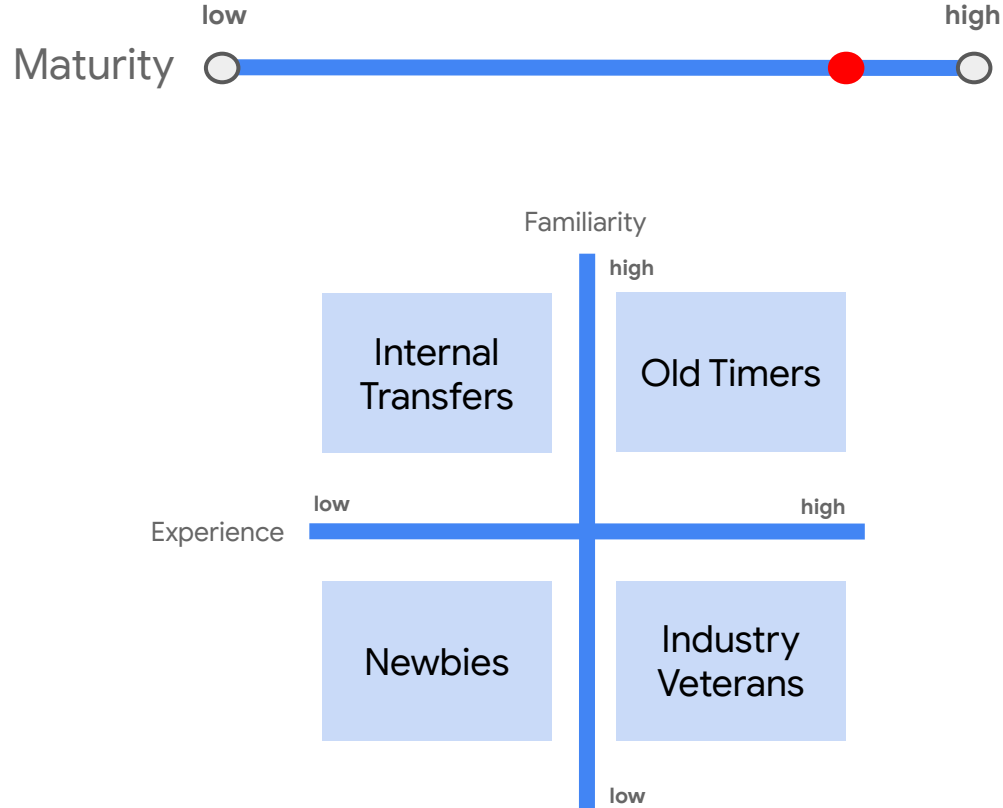
Step 1: Assess the Team Mix



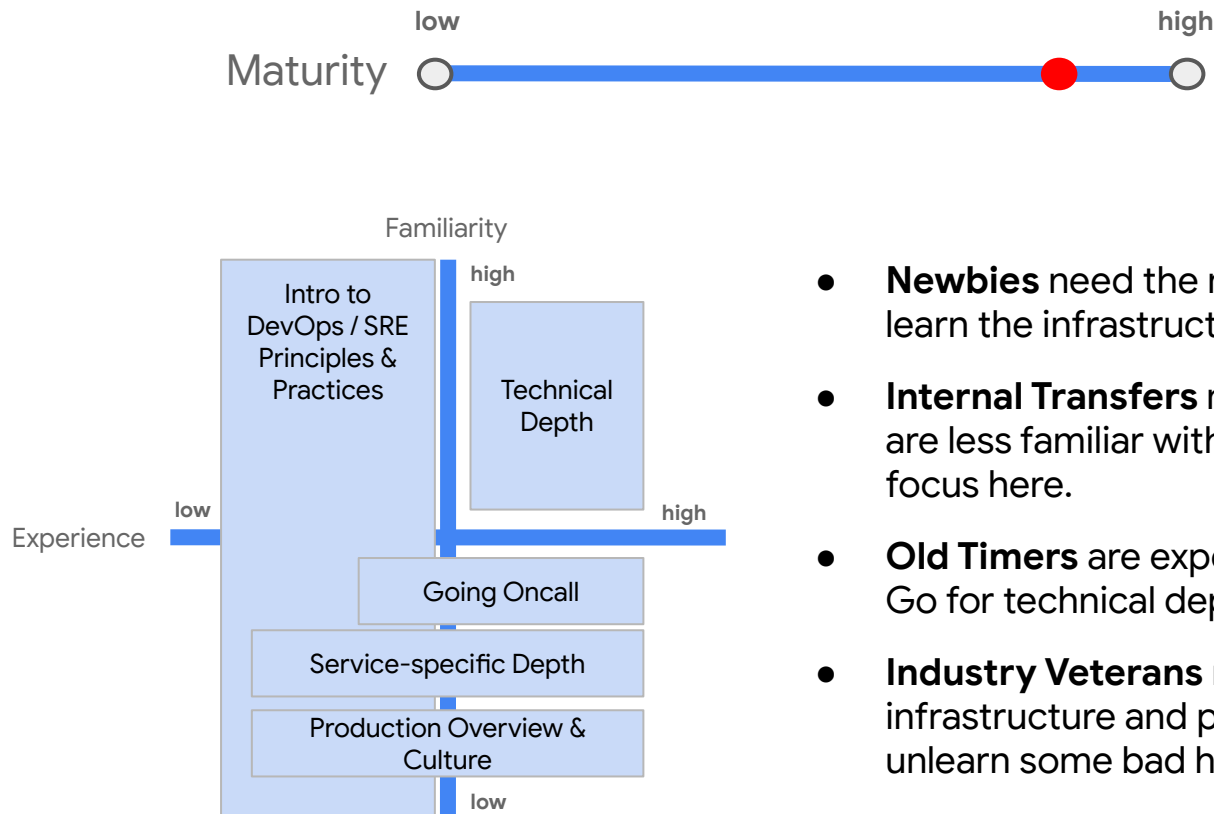
Step 1: Assess the Team Mix



Step 1: Assess the Team Mix



Step 2: Find Your Focus



- **Newbies** need the most attention. They need to learn the infrastructure *and* culture.
- **Internal Transfers** may know your systems but are less familiar with DevOps/SRE practices so focus here.
- **Old Timers** are experienced on both dimensions. Go for technical depth to unlock career growth.
- **Industry Veterans** need to ramp-up on your infrastructure and processes and may need to unlearn some bad habits. Focus on specifics.

WHAT?

Drawing an Analogy Between Software Development and Training Programs

	“What”	“How”
Software Development	Product Features	Deploying to production in a reliable way to meet the needs of our users.
Training Program	Training Content	Deploying a consistent and reliable training program that meets the needs of our students.

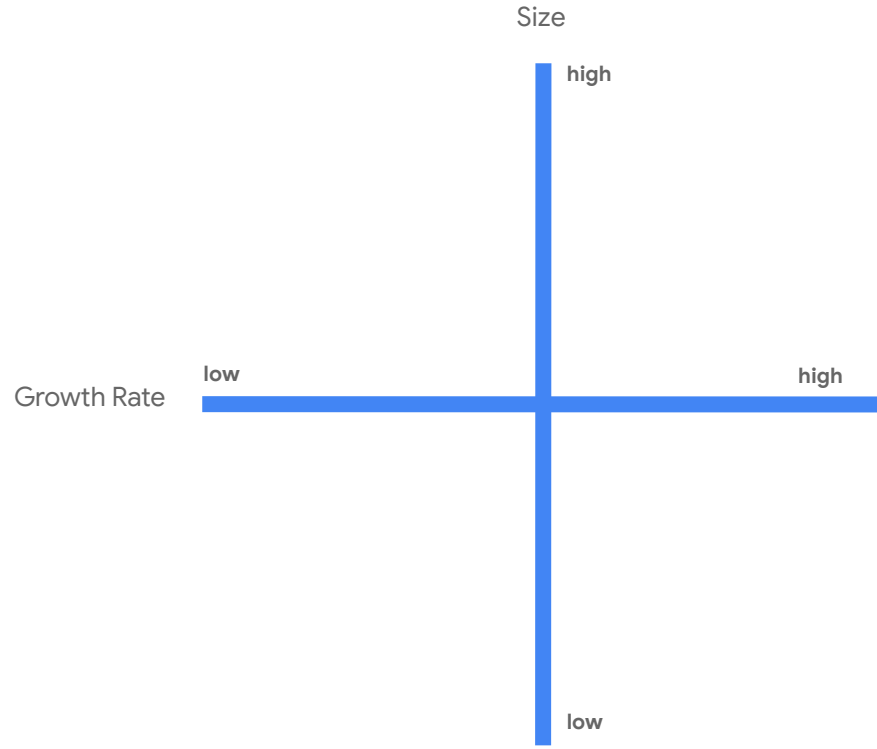
How?

Operational Dimensions Influence Your Training Approach

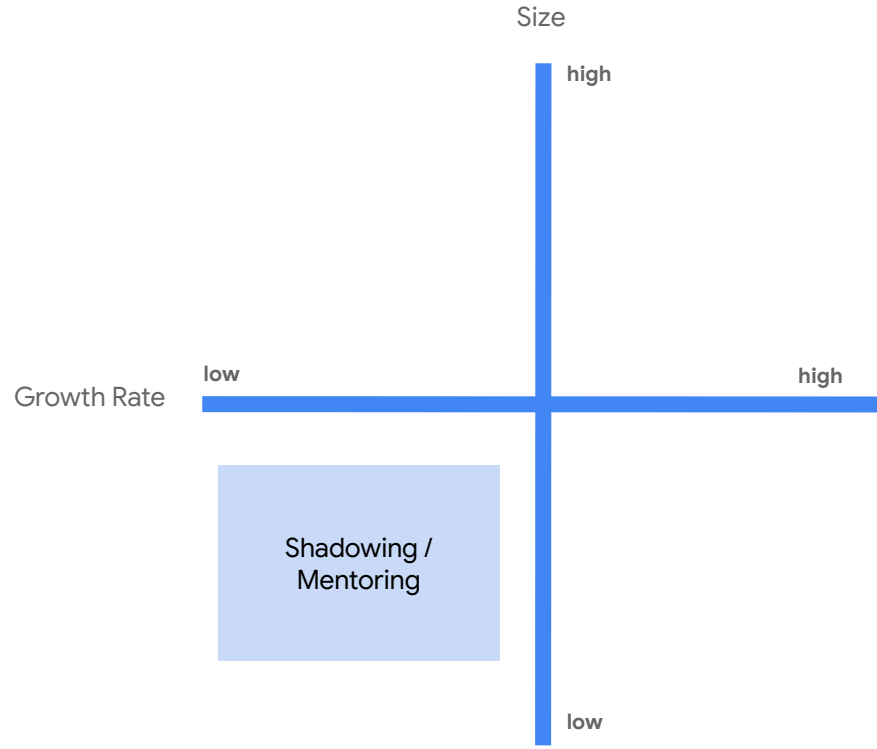
Size  How large is your organization?

Growth Rate  How fast are you growing?

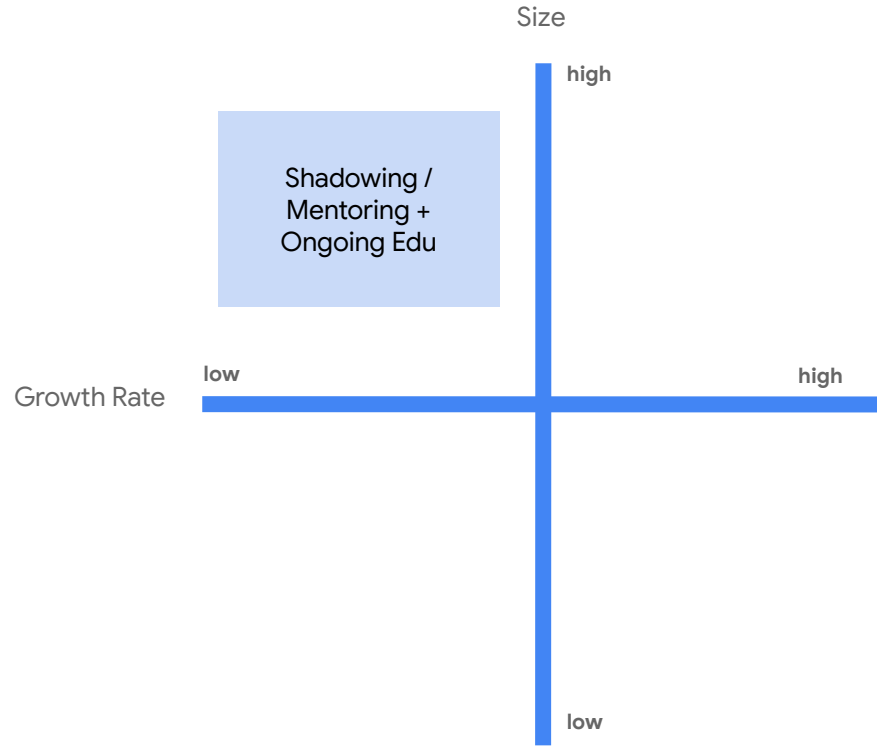
How Much Effort Should You Invest?



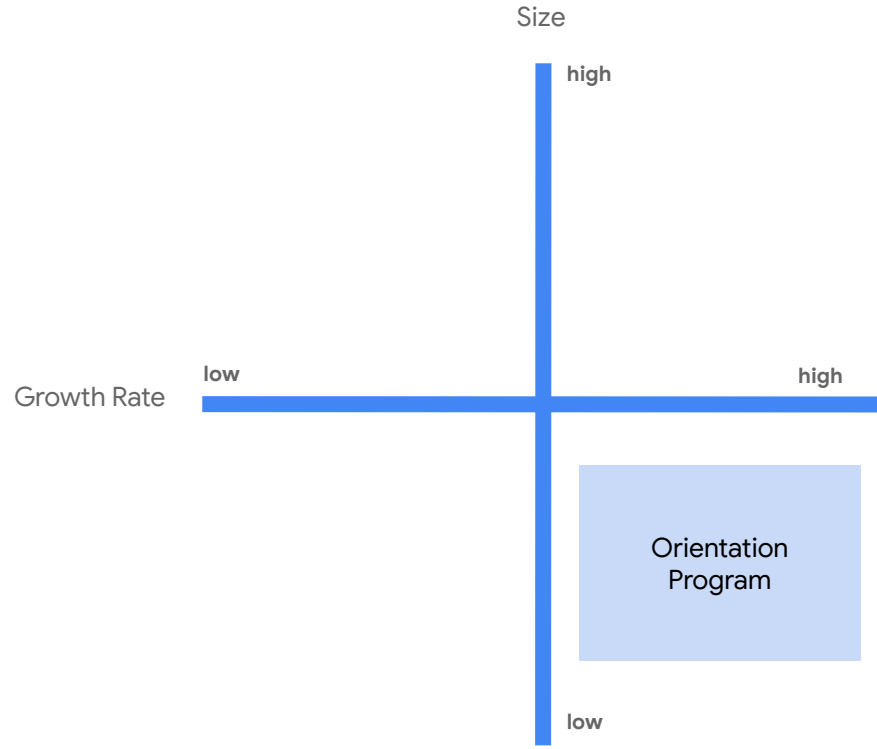
How Much Effort Should You Invest?



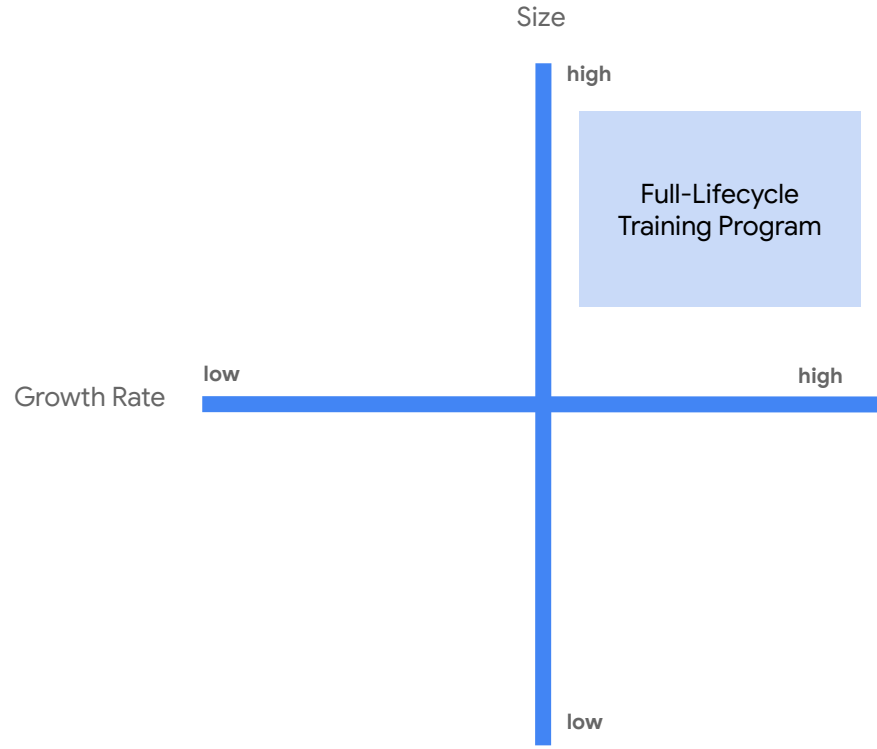
How Much Effort Should You Invest?



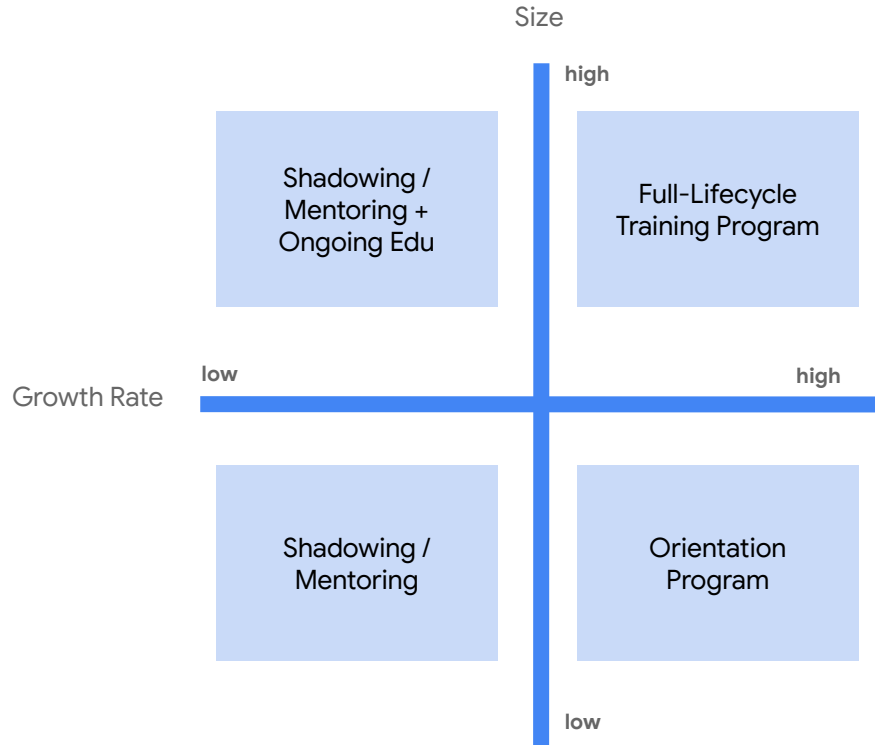
How Much Effort Should You Invest?



How Much Effort Should You Invest?

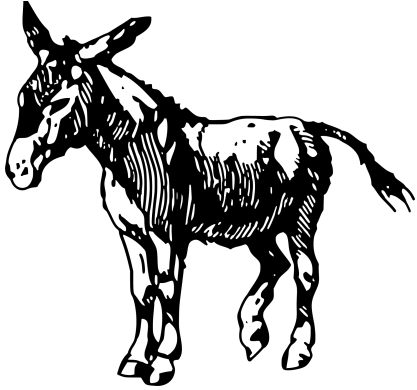


How Much Effort Should You Invest?

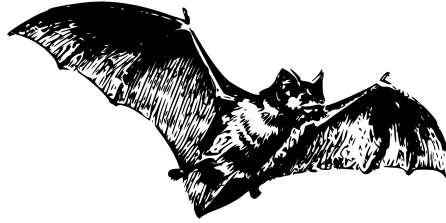


- Invest the most under conditions of rapid growth.
- Small but rapidly growing organizations benefit the most from onboarding training.
- Don't forget about your existing workforce. Invest in ongoing education for their career development.

Where to Start When Developing Content?



+



=

ASSBAT

A **S**tudent **S**hould **B**e **A**ble **T**o

A **S**tudent **S**hould **B**e **A**ble **T**o

Understand \$foo service?

A **S**tudent **S**hould **B**e **A**ble **T**o

~~Understand \$foo service?~~

A Student Should Be Able To

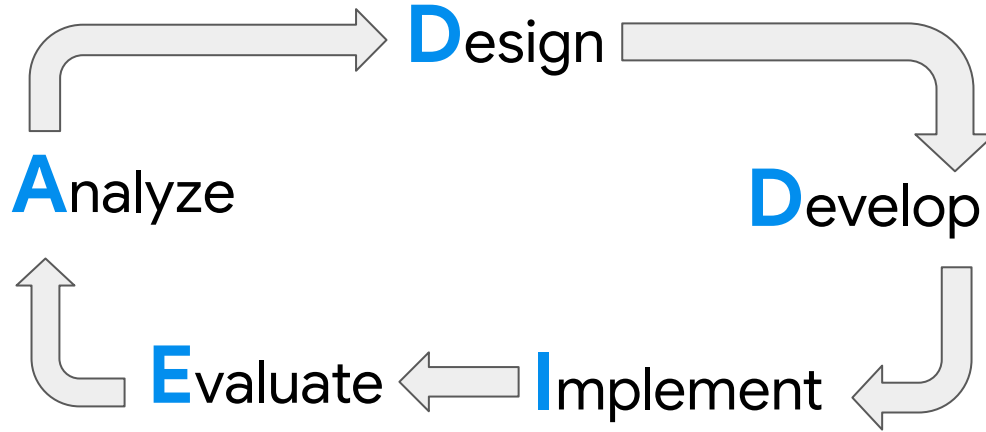
~~Understand \$foo service?~~

Focus on the behaviors you want to drive:

- Use \$tool to identify how much memory a job is using
- Interpret a graph in \$monitoring_tool to identify the health of \$foo service
- Move traffic away from a cluster using \$drain_tool in five minutes

Observe and measure how the training is *applied*

The ADDIE Model of Instructional Design



Analyze needs

Design based on ASSBATS

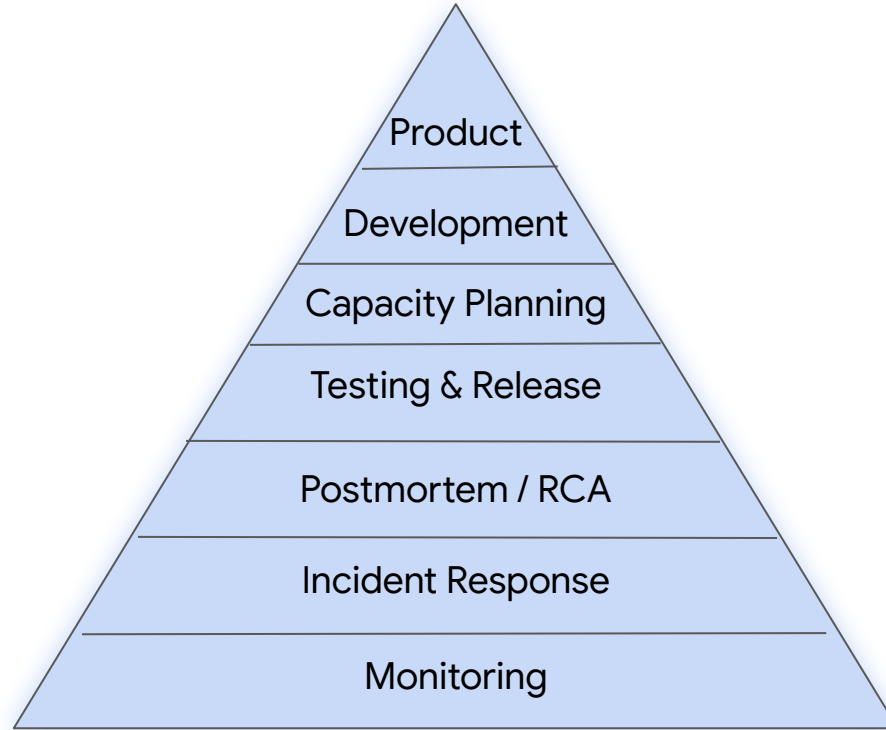
Develop training content

Implement the program

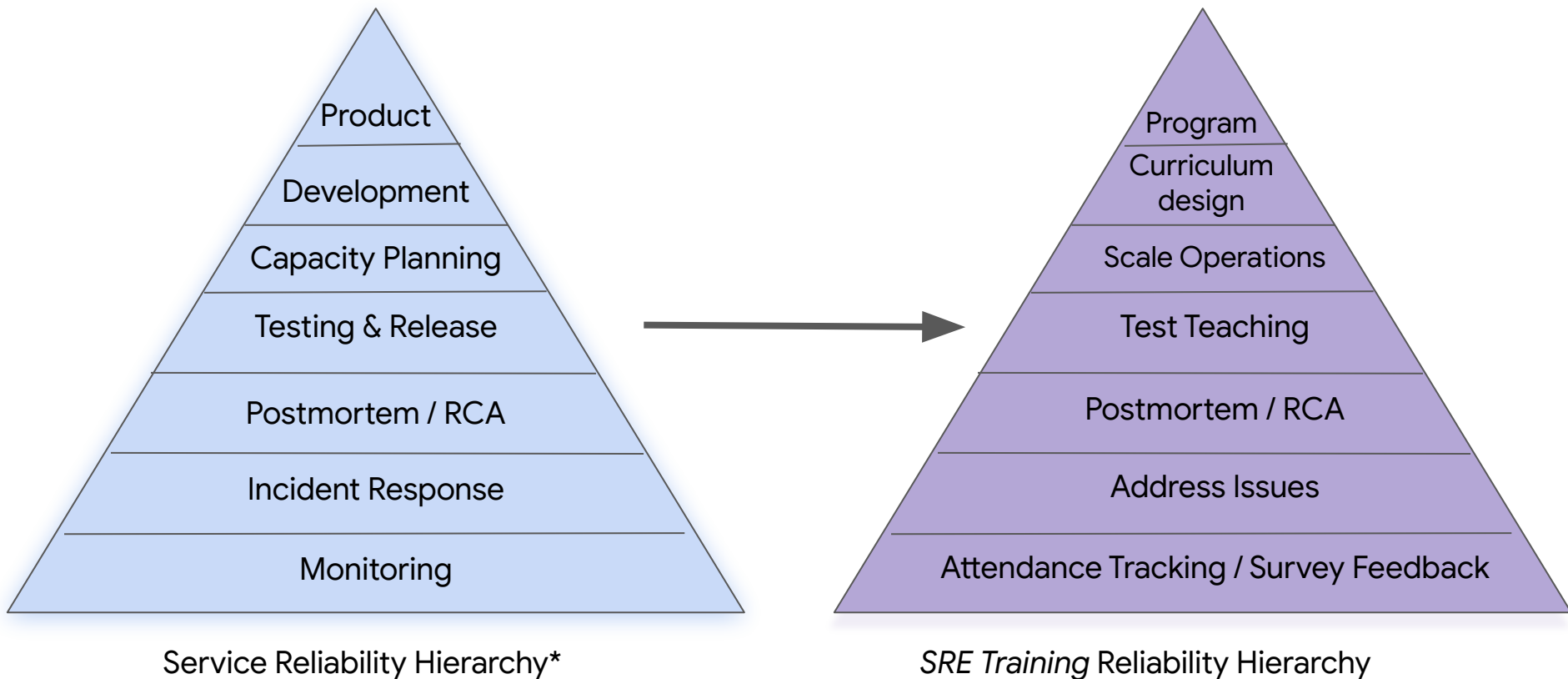
Evaluate effectiveness



Service Reliability Hierarchy



How to Apply SRE Principles to a Training Program



* <https://landing.google.com/sre/sre-book/chapters/part3/>

Is More Effort Always Better? No.



SRE Principle in Practice:

- Do just enough to meet the needs of your students.
- Keep them happy, but not too happy.
- Consider trade-offs and *avoid polishing a diamond.*

What Did Our Monitoring Tell Us?

Also more prepared, **hands-on** "Hello world" demonstrations and in-class labs allowing use of the aforementioned paths would be welcome (kinesthetic).

More time doing **hands-on work** and deeper exploration of how {redacted} were run by SRE teams would be nice.

I disliked the "wall of lecture" in some classes, meaning 1.5 or 2 hours of listening with **little/no hands-on exercise**.

Some more **hands-on activities** would have been good.

Google SRE EDU Orientation v2

- Move away from passive listening
- Troubleshoot a real system, built for this purpose
- Facilitator backs off more and more
- Groups of three students, least experienced in the middle, driving
- Instill confidence



<http://bitly.com/sre-edu>

What Does Our Monitoring Tell Us Now?

I went in feeling quite apprehensive & came out feeling like I at least know which way I'm pointed.
Thoroughly enjoyed the breakage activities and learning about how Google's infra, monitoring and processes fit together.

Delving into real breaking scenarios was super valuable
- I would love more of these (1 per day would be amazing).

The **breakage scenarios in SRE EDU** were **awesome**.

It was the funnest week I've had this year. Overall, it made me feel **more connected to production** and the technology, which made me really happy.

SRE EDU Orientation v2 is Better Instrumented for Observability



Concrete behaviors *demonstrated*

- Use a system diagram
- Diagnose issues using SRE tools
- Annotate an outage
- Mitigate a realistic production issue
- Find root causes & propose solutions

Swim! Don't Sink... Takeaways



Swim! Don't Sink... Takeaways

- **Training is an investment** → An investment in your organization and people.



Swim! Don't Sink... Takeaways

- **Evaluate the cost *and* benefits** → to make sure you make the *right* level of investment.



Swim! Don't Sink... Takeaways

- **Where to invest?** → depends on the 'what' and 'how' of your organizational circumstances.



Swim! Don't Sink... Takeaways

- **Walk the Talk** → Apply SRE / DevOps principles to the training program itself for a consistent and reliable experience.



Swim! Don't Sink... Takeaways

- **Training is an investment** → An investment in your organization and people.
- **Evaluate the cost *and* benefits** → to make sure you make the *right* level of investment.
- **Where to invest?** → depends on the ‘what’ and ‘how’ of your organizational circumstances.
- **Walk the Talk** → Apply SRE / DevOps principles to the training program itself for a consistent and reliable experience.



Time for a Final Poll...



Thanks to the SRE EDU Core Team and All Our Volunteers!

Laura Baum

Program Manager

Benjamin Weaver

Program Mgr



Brad Lipinski

SRE, Software
Engineer

Jennifer Petoff

Global Program Mgr &
Lead

David Butts

SRE, Software
Engineer

JC van Winkel

Lead Educator

Preston Yoshioka

Instructional
Designer

Want to learn more?

<http://bitly.com/training-sres>



Jennifer Petoff (aka Dr. J)
Twitter: @jenski
Global Head of SRE EDU at Google