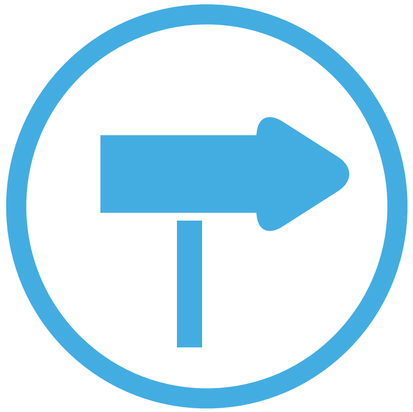


Tech Lead Masterclass

From Maker to Multiplier

Patrick Kua
20+ years experience

Agile Software Development

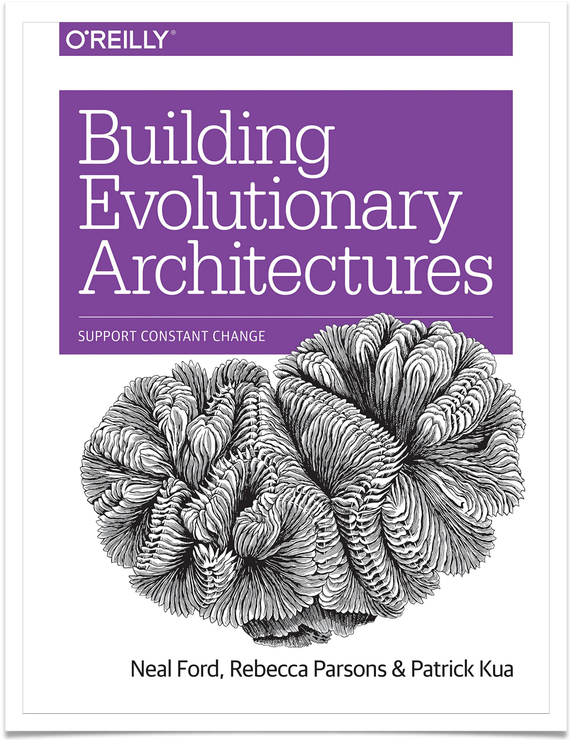
Organisational Change

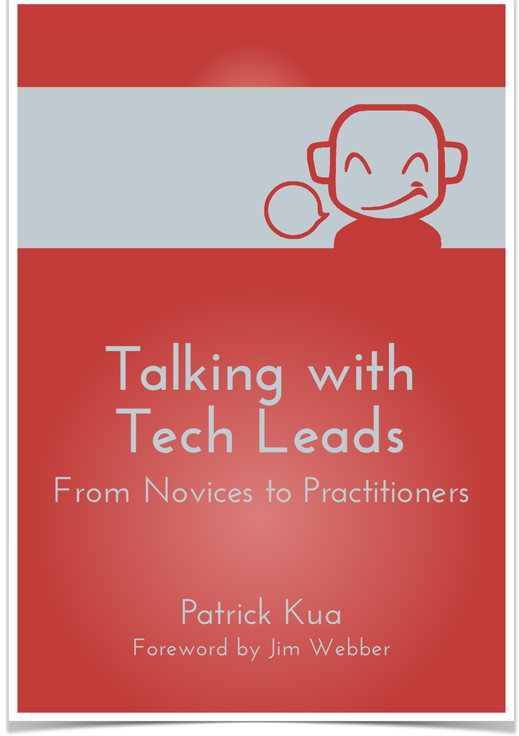
Systems Thinking

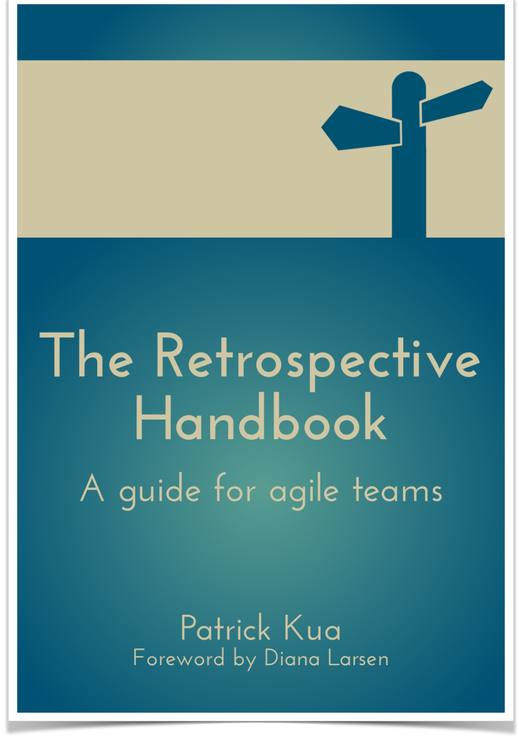
Leadership Development

#Architect #Developer #Coach #Leader #CTO

#Life-long learner #Author #Speaker







thekua.io/evolarch thekua.io/twtl thekua.io/retrobook

#Architect #Developer #Coach #Leader #CTO

#Life-long learner #Author #Speaker

Level Up

A curated newsletter for leaders in tech such as Tech Leads, Engineering Managers, VPs Engineering & CTOs

http://levelup.patkua.com

Logistics

- Timing
- Breaks
- Amenities
- Resources + Note Taking
- Distributed notes
- What we cover

Agenda

AGENDA

Introductions

Hopes and Concerns

Tech Lead Models

Maker vs Multipler Mindset

Programming

Cross Functional Requirements + Technical Vision

People

Process

Hopes & Concerns Recap

Action Plan

How we are going to work

Mixture of Styles

How we are going to work

Lecture

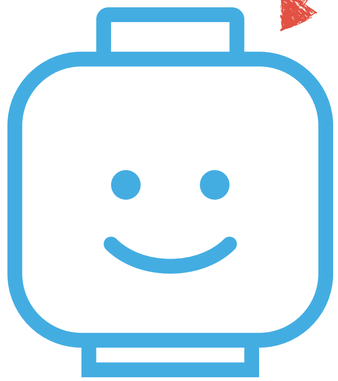
Discussion

Hands on Exercises

Learning from each other

EXERCISE

Draw a picture of your partner



Name

Background:

Small batch XOXO tousled tacos stumptown. Pickled raclette YOLO put a bird on it. Beard vinyl deep v retro PBR&B green juice craft beer hammock everyday carry put a bird on it. Poutine butcher shoreditch food truck organic, gentrify unicorn plaid semiotics truffaut. Microdosing la croix banjo, prism next level quinoa deep v pabst single-origin coffee.

Tech Lead Superpower:

Seitan kombucha tofu making

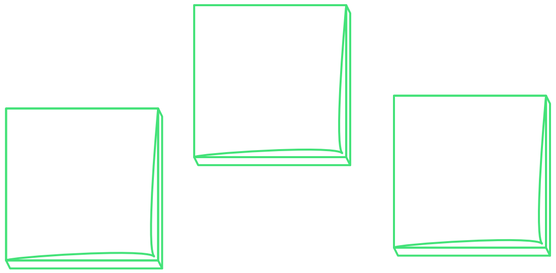
Favourite Beverage:

Intelligentsia small batch cold brew

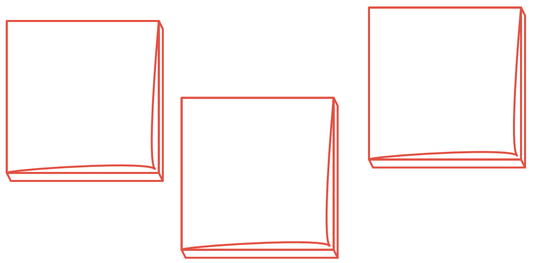
EXERCISE

Hopes & Concerns

What is your **biggest hope** today?

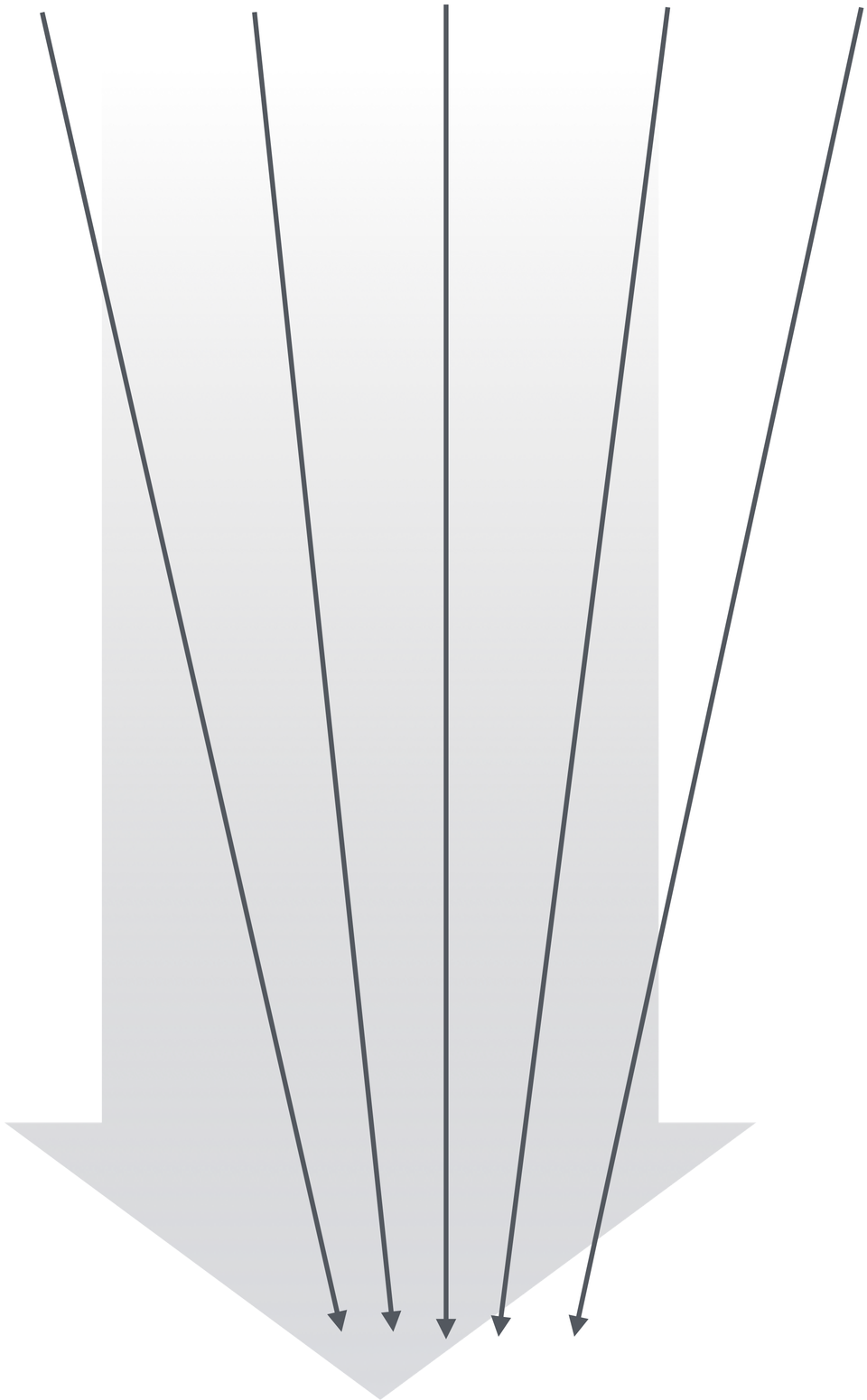


What **is your biggest concern** today (about the Tech Lead role)?

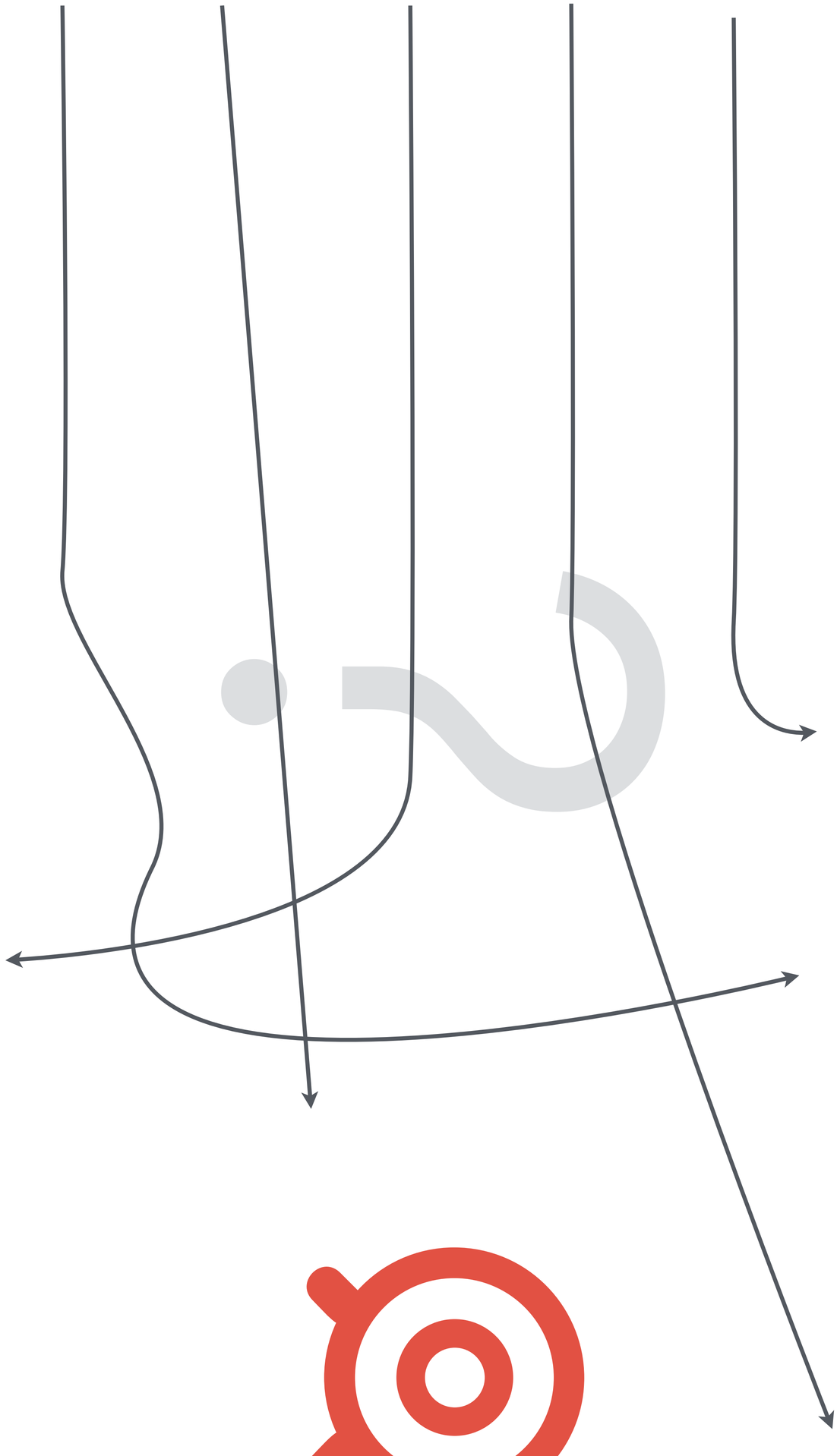


Why do we need a Tech Lead?

Do Do Do Do Do



Do Do Do Do Do



http://www.flickr.com/photos/dcarlbom/3468358859/

Think this doesn’t happen in the real world?

@julianboot

@thejayfields: I had ten guys on my last project, all of them had opinions and all of them were expressed in the code base #speakerconf

Source: http://twitter.com/julianboot/status/232830267822309376

Tech Lead Models

TECH LEAD

Is...

a role

a developer who leads a team

Could be...

a Lead Developer

an Architect who codes

Is not...

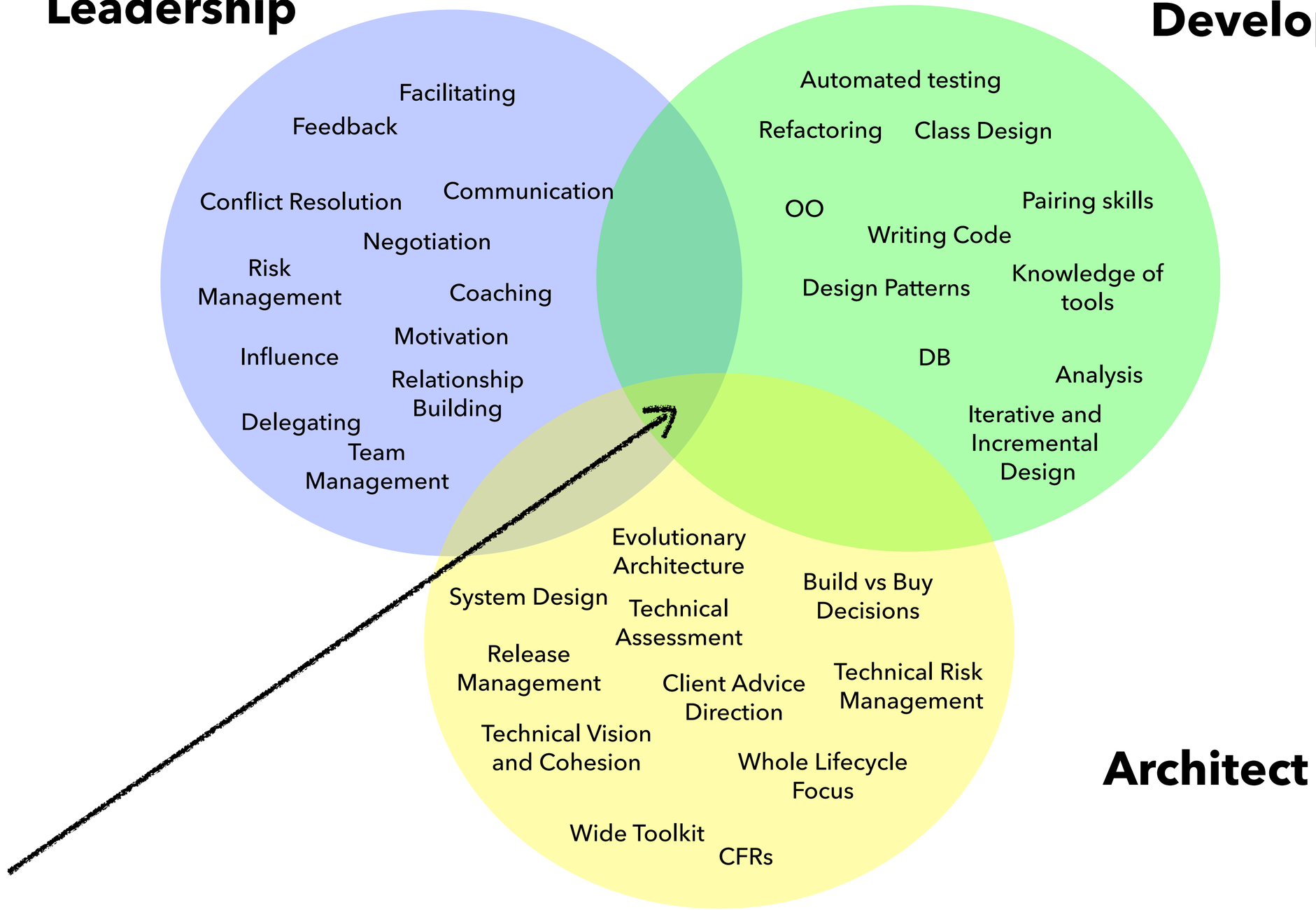
(just) the team leader

the Scrum Master

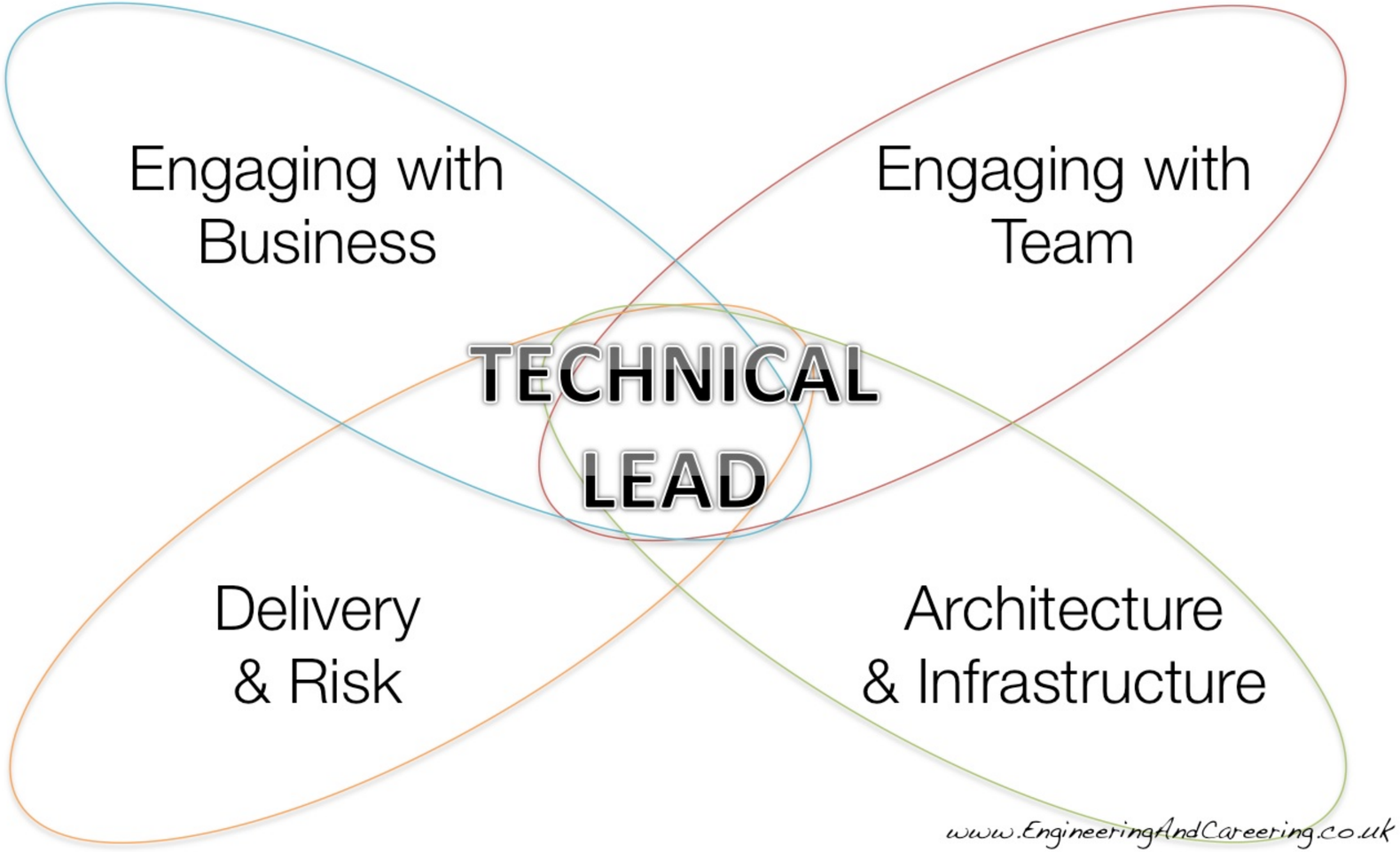
a Technical Manager

always the best technical person

TECH LEAD


Leadership
Developer
Architect
Facilitating
Feedback
Conflict Resolution
Communication
Negotiation
Risk Management
Coaching
Motivation
Influence
Relationship Building
Delegating
Team Management
Automated testing
Refactoring
Class Design
OO
Pairing skills
Writing Code
Design Patterns
Knowledge of tools
DB
Analysis
Iterative and Incremental Design
Evolutionary Architecture
System Design
Technical Assessment
Build vs Buy Decisions
Release Management
Client Advice Direction
Technical Risk Management
Technical Vision and Cohesion
Whole Lifecycle Focus
Wide Toolkit
CFRs

Dan Abel's Well-Rounded Tech Lead model



Source: http://www.engineeringandcareering.co.uk/2013/05/the-well-rounded-technical-lead-model.html

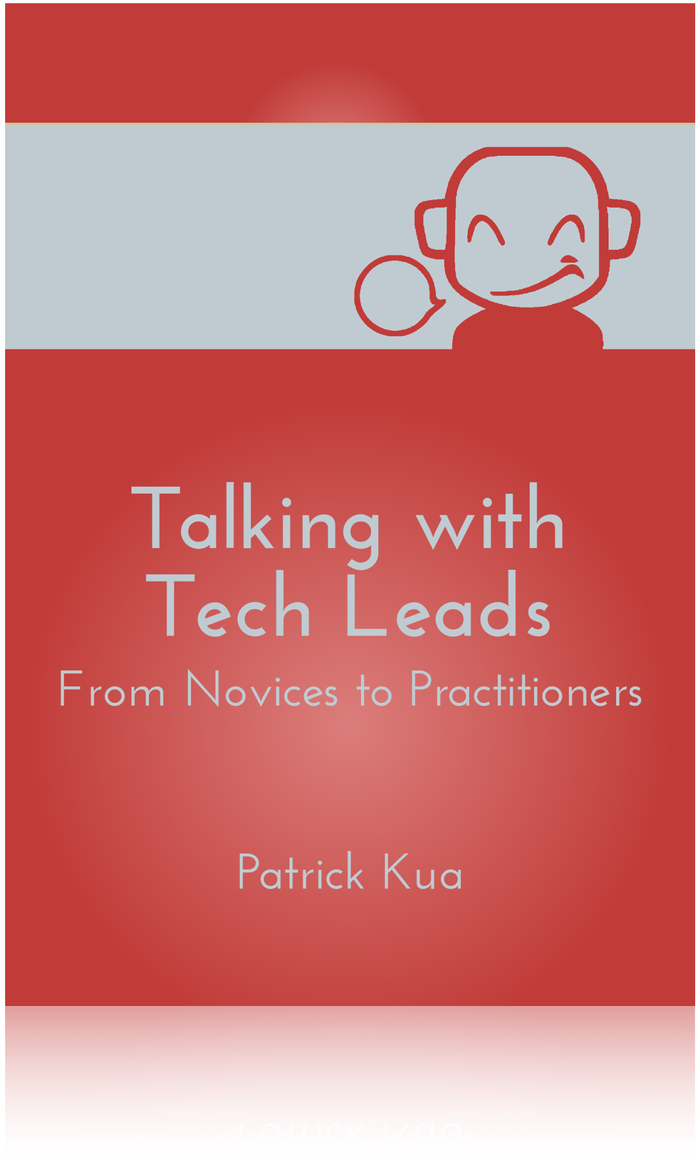
TECH LEAD

Bridging the Business with Tech

"Tech" of being a Tech Lead

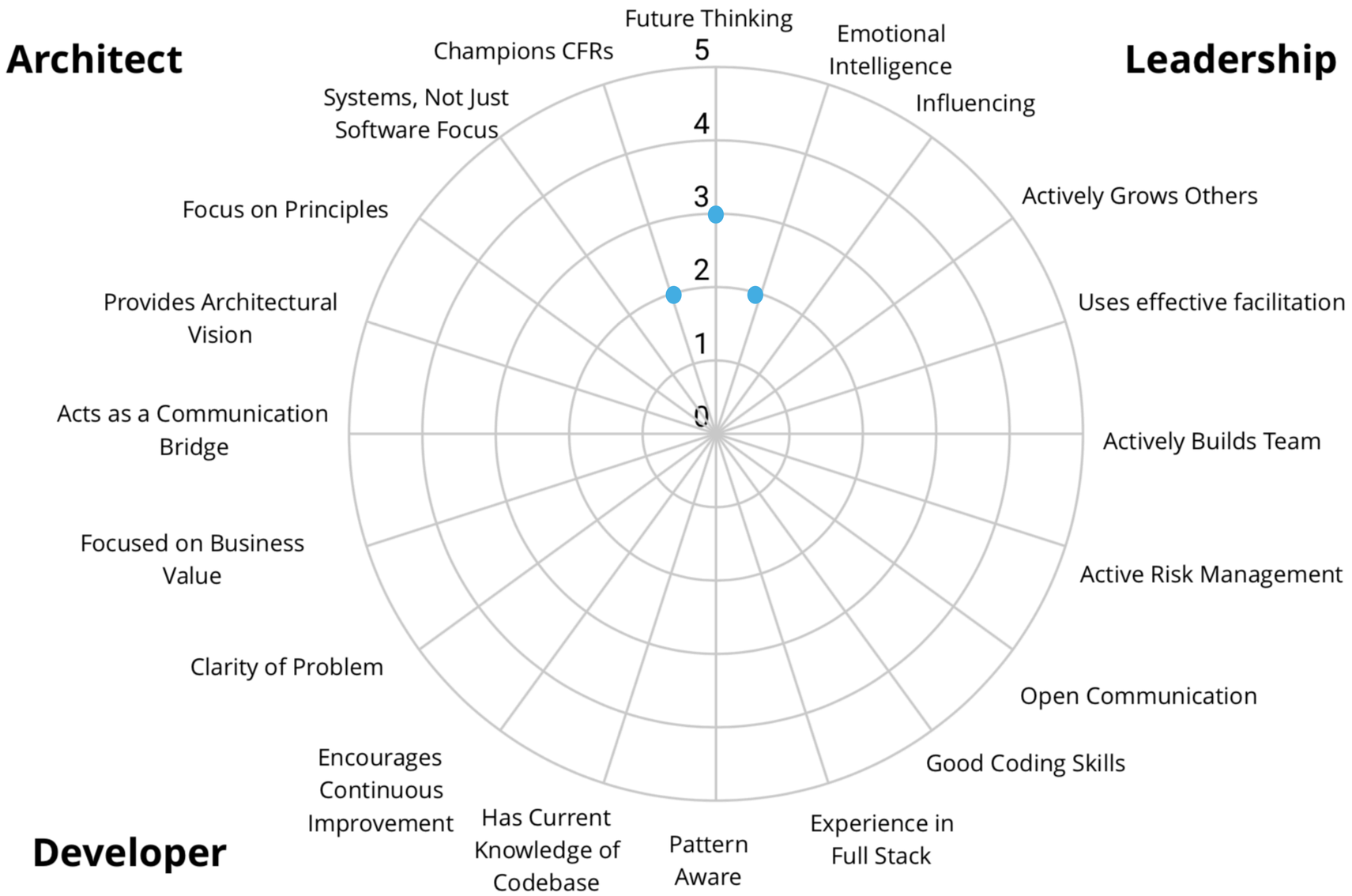
People

You

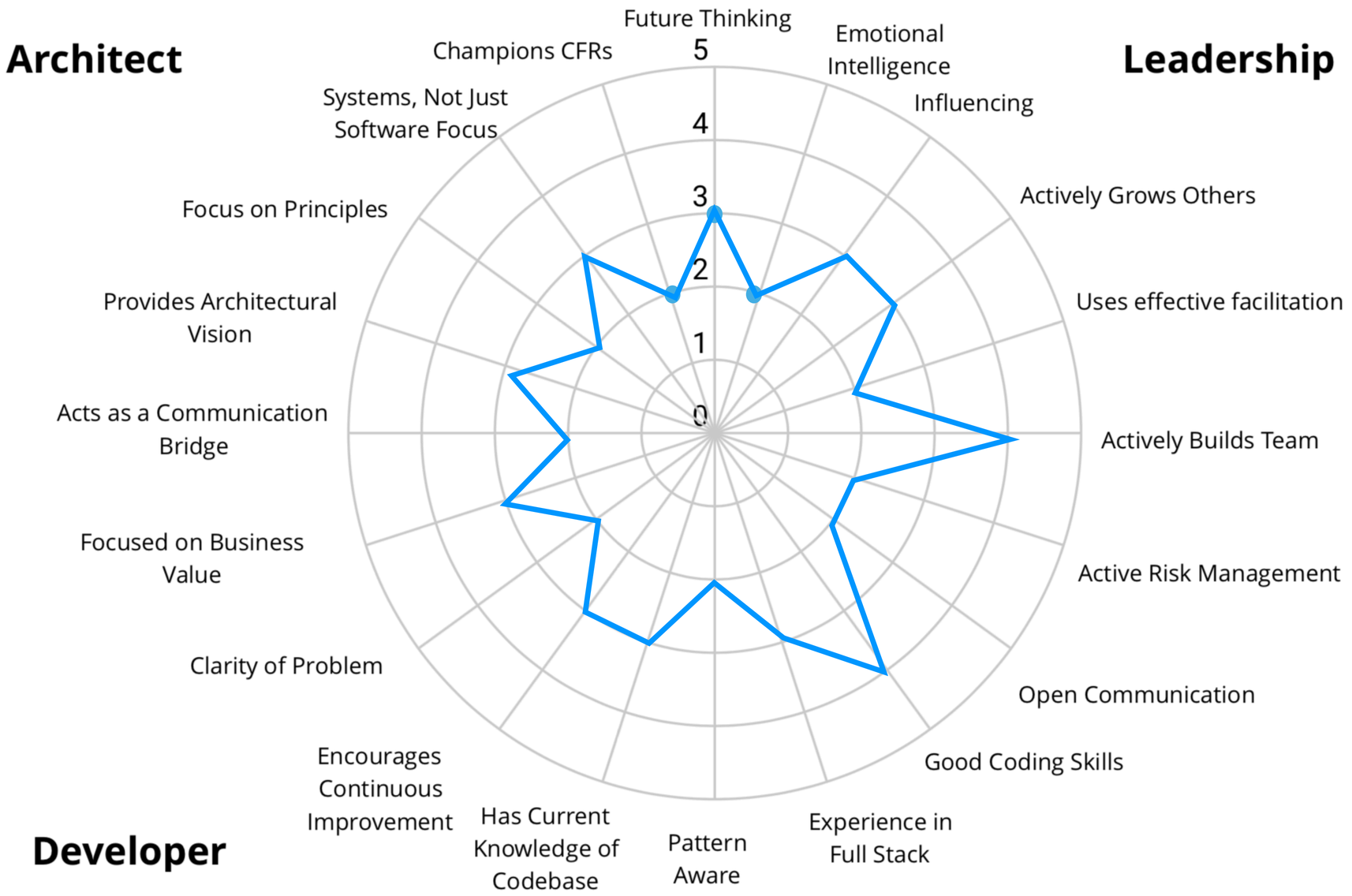


Tech Lead Self-Assessment

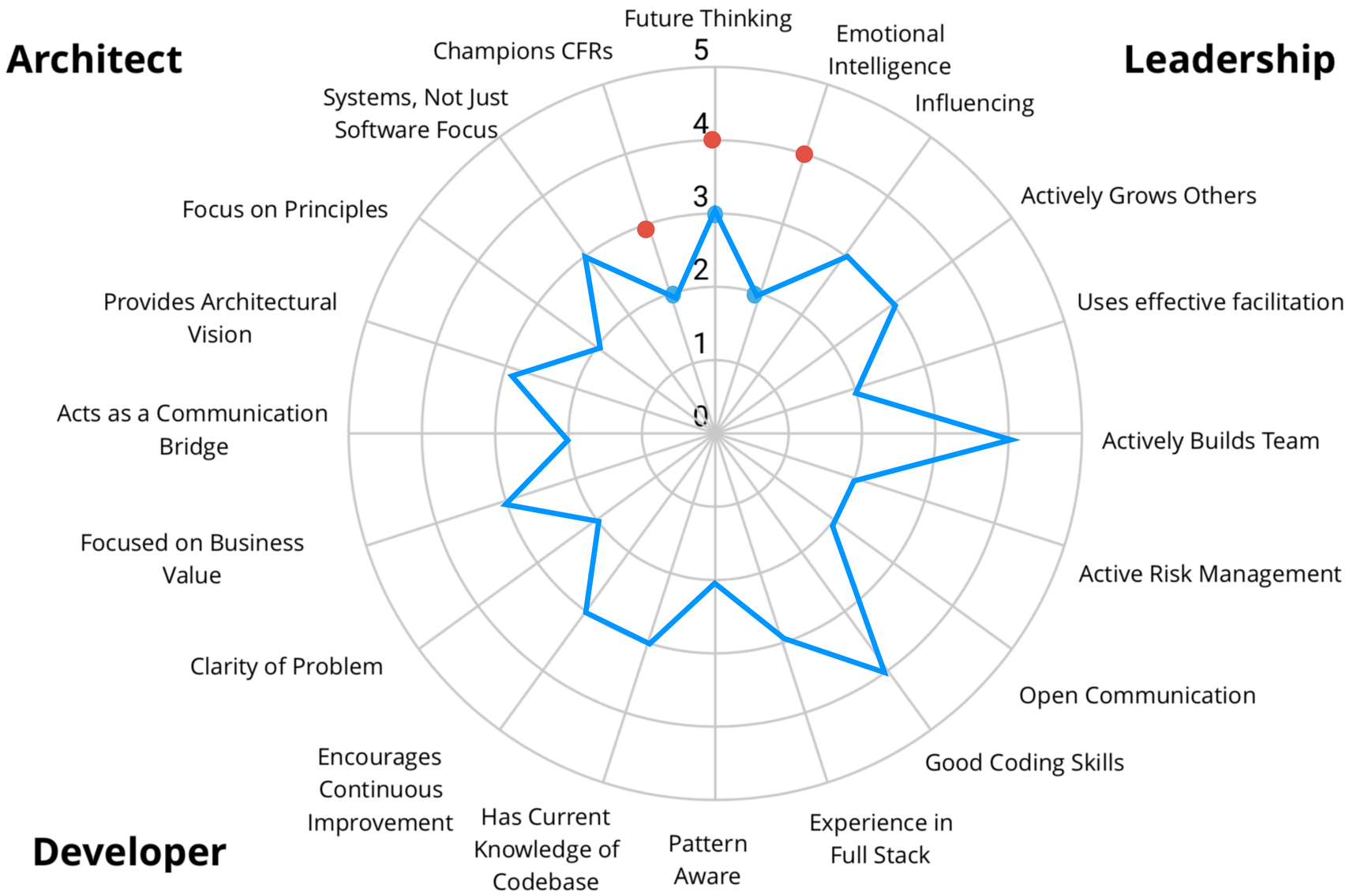
EXERCISE


Architect
Leadership
Developer
Future Thinking
Emotional Intelligence
Influencing
Actively Grows Others
Uses effective facilitation
Actively Builds Team
Active Risk Management
Open Communication
Good Coding Skills
Experience in Full Stack
Pattern Aware
Has Current Knowledge of Codebase
Encourages Continuous Improvement
Clarity of Problem
Focused on Business Value
Acts as a Communication Bridge
Provides Architectural Vision
Focus on Principles
Systems, Not Just Software Focus
Champions CFRs
5
4
3
2
1
0

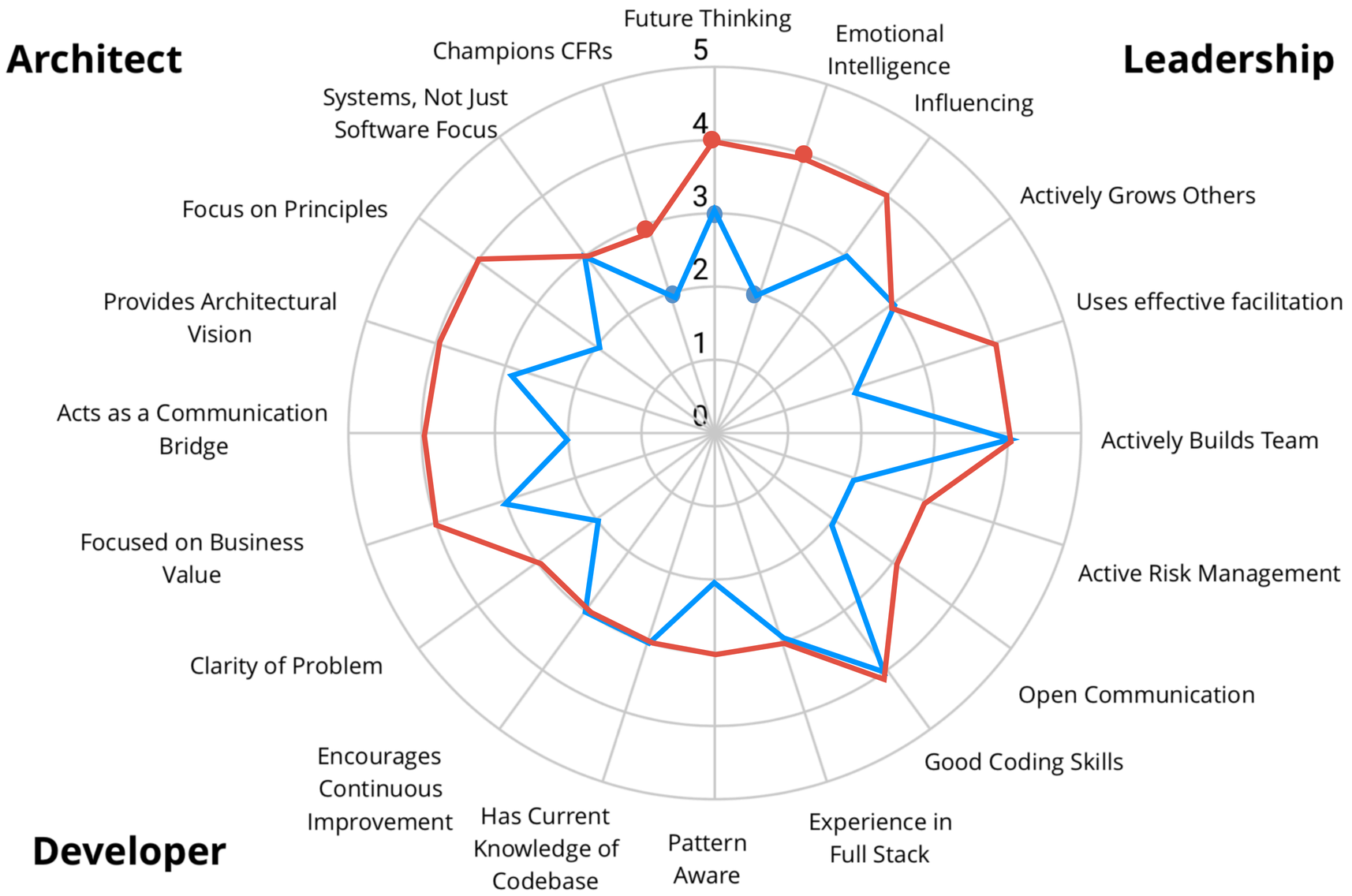
EXERCISE


Architect
Leadership
Developer
Future Thinking
Emotional Intelligence
Influencing
Actively Grows Others
Uses effective facilitation
Actively Builds Team
Active Risk Management
Open Communication
Good Coding Skills
Experience in Full Stack
Pattern Aware
Has Current Knowledge of Codebase
Encourages Continuous Improvement
Clarity of Problem
Focused on Business Value
Acts as a Communication Bridge
Provides Architectural Vision
Focus on Principles
Systems, Not Just Software Focus
Champions CFRs
5
4
3
2
1
0

EXERCISE


Architect
Leadership
Developer
Future Thinking
Emotional Intelligence
Influencing
Actively Grows Others
Uses effective facilitation
Actively Builds Team
Active Risk Management
Open Communication
Good Coding Skills
Experience in Full Stack
Pattern Aware
Has Current Knowledge of Codebase
Encourages Continuous Improvement
Clarity of Problem
Focused on Business Value
Acts as a Communication Bridge
Provides Architectural Vision
Focus on Principles
Systems, Not Just Software Focus
Champions CFRs
5
4
3
2
1
0

EXERCISE


Architect
Leadership
Developer
Future Thinking
Emotional Intelligence
Influencing
Actively Grows Others
Uses effective facilitation
Actively Builds Team
Active Risk Management
Open Communication
Good Coding Skills
Experience in Full Stack
Pattern Aware
Has Current Knowledge of Codebase
Encourages Continuous Improvement
Clarity of Problem
Focused on Business Value
Acts as a Communication Bridge
Provides Architectural Vision
Focus on Principles
Systems, Not Just Software Focus
Champions CFRs
5
4
3
2
1
0

EXERCISE

What does good look like?

What not to do

A story of poor technical leadership

"The Overnight Refactor"

EXERCISE

What does good look like?

Good Behaviours

Bad Behaviours

In your groups, brainstorm behaviours of what makes a good / bad technical leader (1 item per sticky note, use as many as you like) and discuss these


Chrome File Edit View History Bookmarks People Window Help
Deviation
master
src/components/Player.jsx
I am a MainContent
Authentication failed
I am a Player

Maker

Maker to Multiplier

SIMPLE TESTS...

The Conceptual Integrity Test

Does the codebase look like it was written by a single person?

☐ *Yes* ☐ *No*

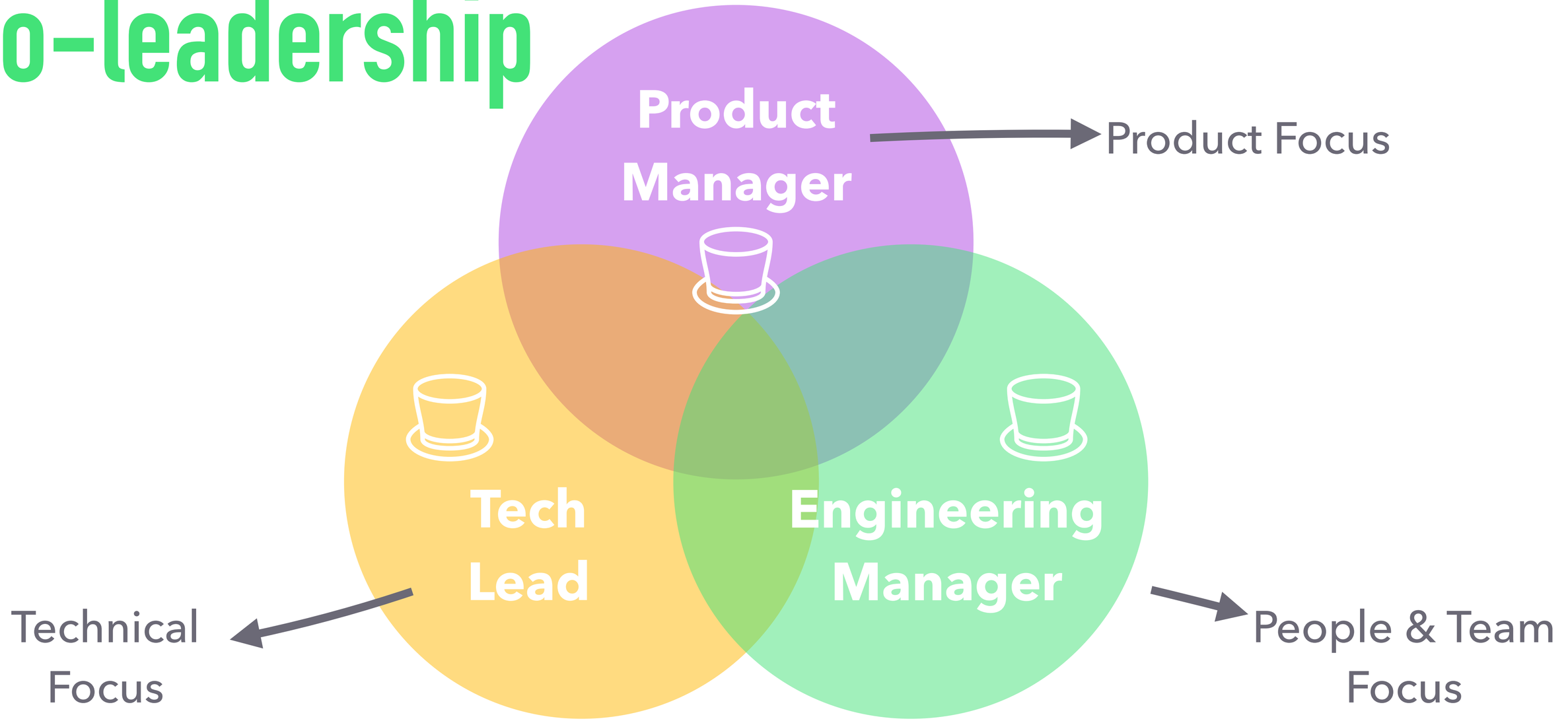
The Holiday Test

Can you go on holiday for 2 weeks?

(without responding to email/slack *and*
everything continuing to work well)

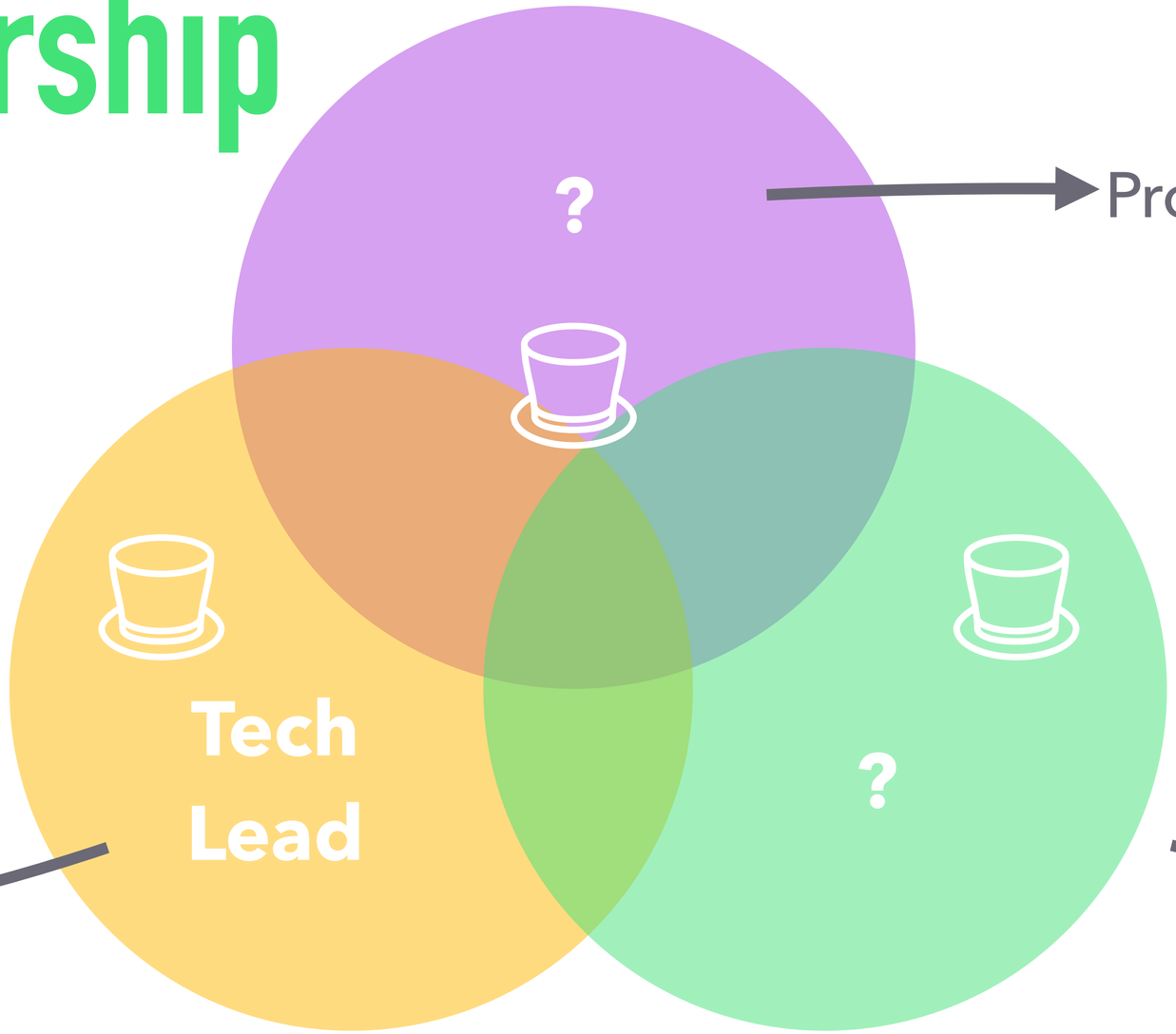
☐ *Yes* ☐ *No*

Co-leadership



Key Responsibilities

Co-leadership



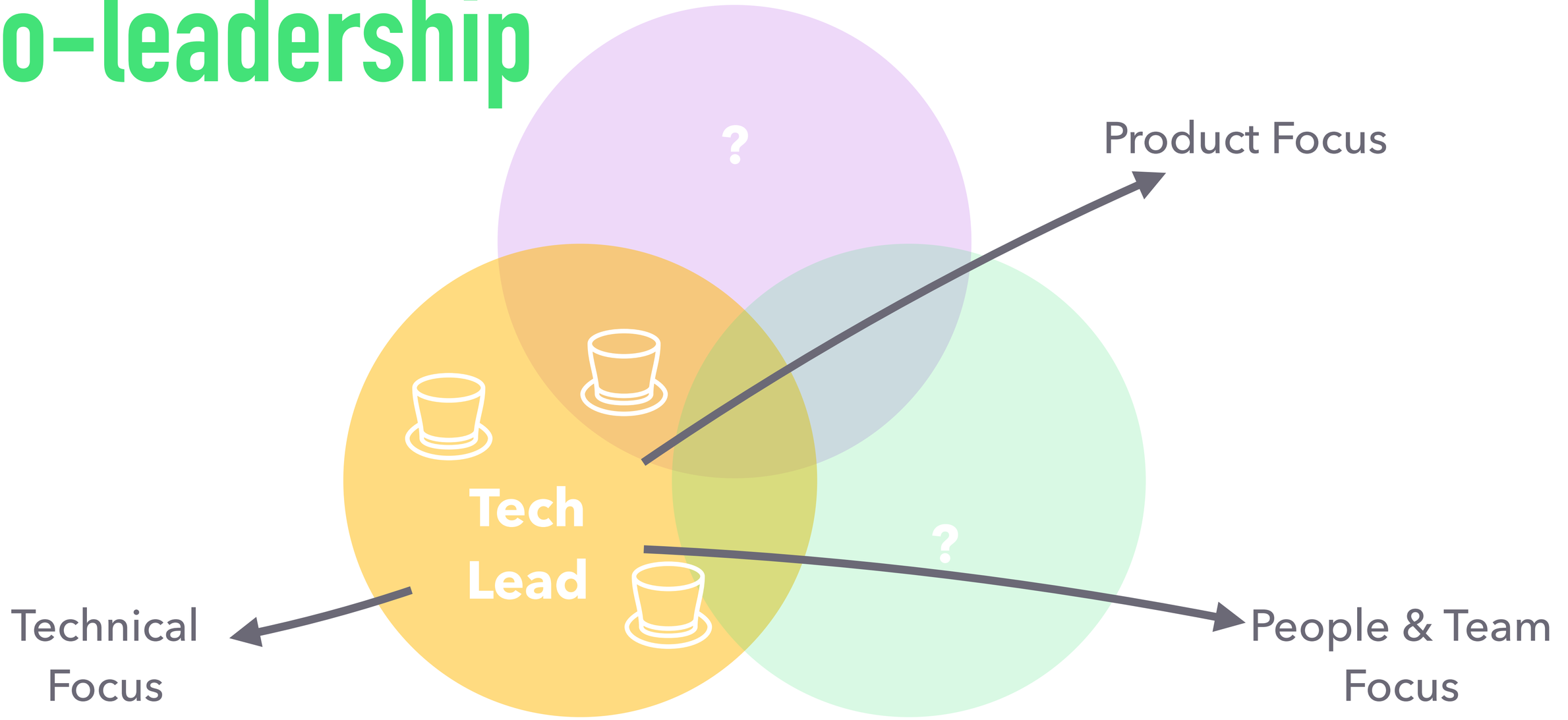
Product Focus

Technical
Focus

People & Team
Focus

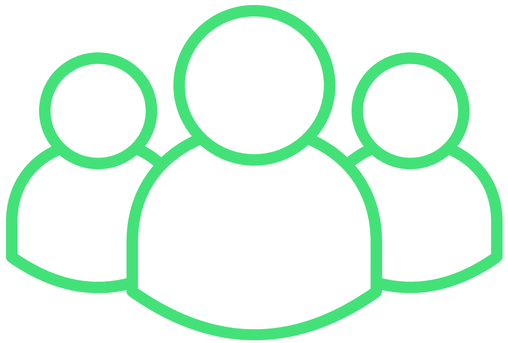
Key Responsibilities

Co-leadership

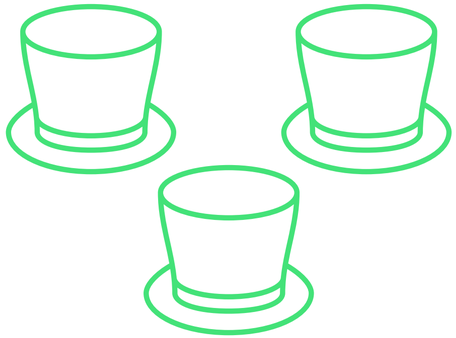


Key Responsibilities

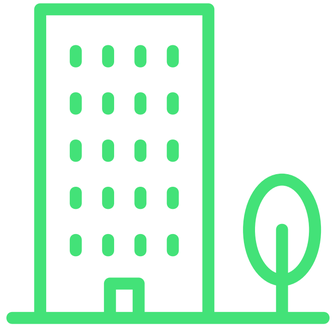
Responsibilities vary



Team Size



Roles in Team



Organisation

WHAT DOES A GOOD TECH LEAD FOCUS ON?

Programming

People

Process

Programming

DEFINITELY

?

Do effective Tech Leads need to code?

At least **30%** *of the time with the team*

"…respect is the currency of the realm"

COMPUTERWORLD

Print Article Close Window

Opinion: The unspoken truth about managing geeks

j.ello

September 8, 2009 (Computerworld)

I can sum up every article, book and column written by notable management experts about managing IT in two sentences: "Geeks are smart and creative, but they are also egocentric, antisocial, managerially and business-challenged, victim-prone, bullheaded and credit-whoring. To overcome these intractable behavioral deficits you must do X, Y and Z."

X, Y and Z are variable and usually contradictory between one expert and the next, but the patronizing stereotypes remain constant. I'm not entirely sure that is helpful. So, using the familiar brush, allow me to paint a different picture of those IT pros buried somewhere in your organization.

Jeff Ello

My career has been stippled with a good bit of disaster recovery consulting, which has led me to deal with dozens of organizations on their worst day, when opinions were pretty raw. I've heard all of the above-mentioned stereotypes and far worse, as well as good bit of rage. The worse shape an organization is in, the more you hear the stereotypes thrown around. But my personal experiences working within IT groups have always been quite good, working with IT pros for whom the negative stereotypes just don't seem to apply. I tended to chalk up IT group failures to some bad luck in hiring and the delicate balance of those geek stereotypes.

Recently, though, I have come to realize that perfectly healthy groups with solid, well-adjusted IT pros can and will devolve, slowly and quietly, into the behaviors that give rise to the stereotypes, given the right set of conditions. It turns out that it is the conditions that are stereotypical, and the IT pros tend to react to those conditions in logical ways. To say it a different way, organizations actively elicit these stereotypical negative behaviors.

Understanding why IT pros appear to act the way they do makes working with, among and as one of them the easiest job in the world.

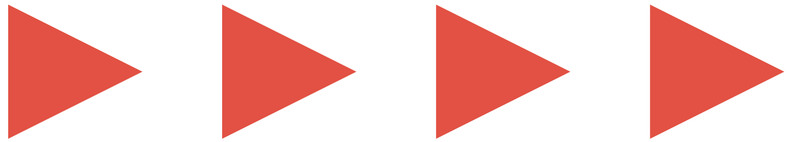
It's all about respect

Few people notice this, but for IT groups respect is the currency of the realm. IT pros do not squander this currency. Those whom they do not believe are worthy of their respect might instead be treated to professional courtesy, a friendly demeanor or the acceptance of authority. Gaining respect is not a matter of being the boss and has nothing to do with being likeable or sociable; whether you talk, eat or smell right; or any measure that isn't directly related to the work. The amount of respect an IT pro pays someone is a measure of how tolerable that person

WHY?

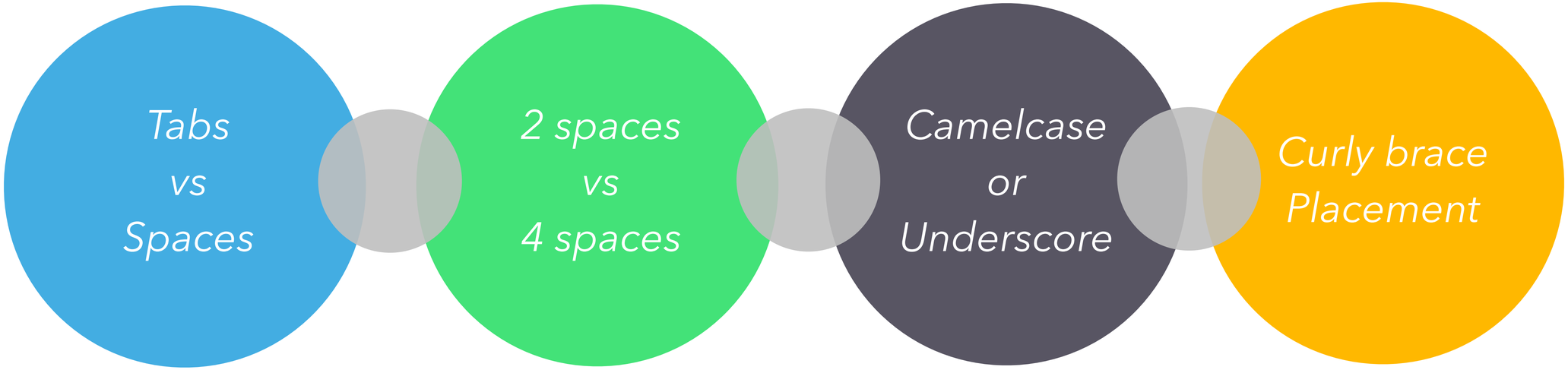
http://bit.ly/15Rm4z

Consistency **over** cleverness



FOCUS

No arguments over...



There are more important topics to spend time on

Reducing cognitive load for the team

Essential complexity

Accidental complexity

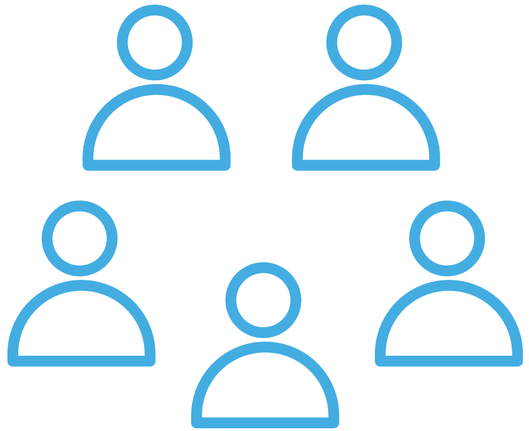
What are some examples?

FOCUS

Software Quality

What sort of quality do you have?

What feedback loops do you have?



What are the speed of those feedback loops?

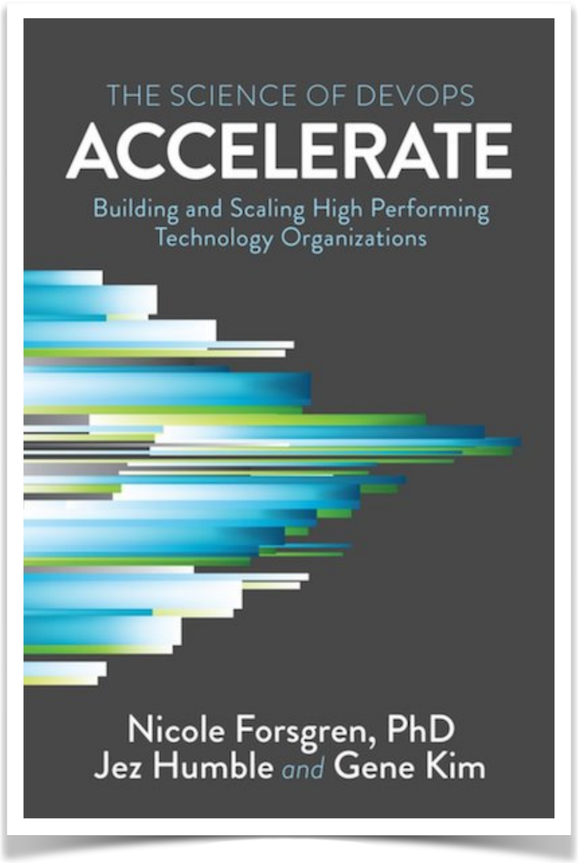
EXERCISE

Brainstorm

Use a Flipchart, and brainstorm activities you can do as a technical leader to improve technical quality?

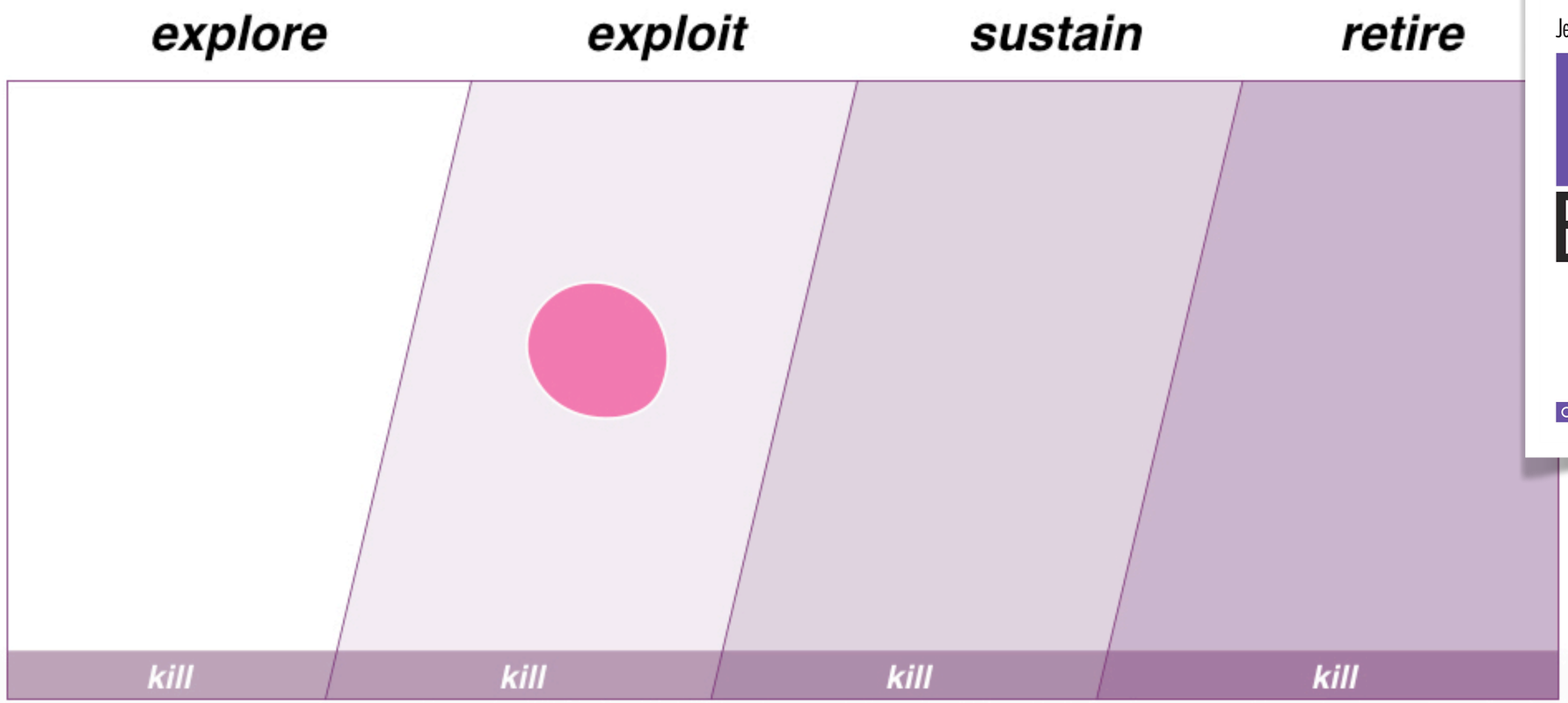
Research Shows Quality Matters

4 Key Metrics

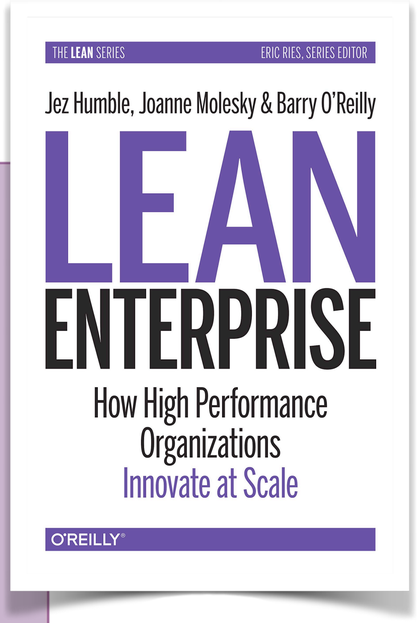


- Lead time
- Deploy frequency
- Mean Time to Restore (MTTR)
- Change fail percentage

Quality is contextual



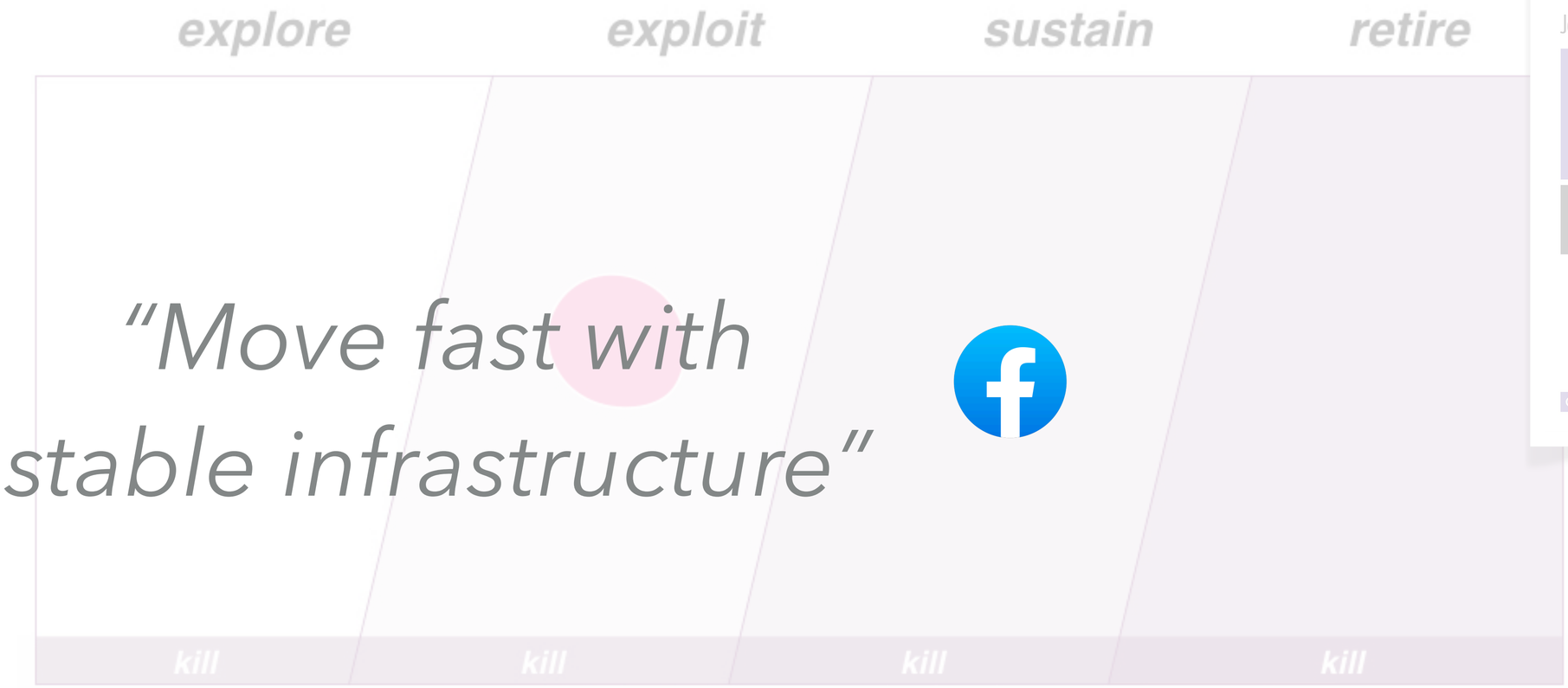
Humble, Molesky, O'Reilly, *Lean Enterprise: How High Performance Organizations Innovate At Scale*



Quality is contextual

Humble, Molesky, O'Reilly, *Lean Enterprise: How High Performance Organizations Innovate At Scale*

Quality is contextual



"Move fast with stable infrastructure"

Humble, Molesky, O'Reilly, *Lean Enterprise: How High Performance Organizations Innovate At Scale*

CROSS FUNCTIONAL REQUIREMENTS

"What sort of quality do you want?"

What are Cross Functional Requirements (CFRs)

Sometimes called:

- Quality attributes
- “-ilities”
- “Non-Functional Requirements”
- The qualities no one generally asks about

What are Cross Functional Requirements (CFRs)

These “cut across” all functional requirements

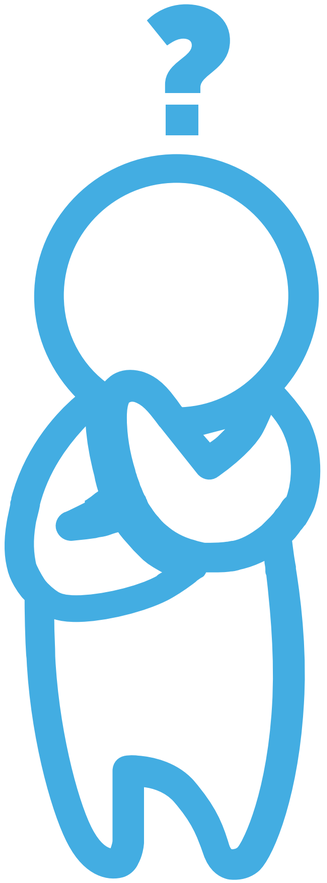
- “Make or break” a system
- Significantly shape architecture
- Impacts tooling/process
- Can have significant cost

Why avoid “Non-Functional Requirements (NFRs)”?

People (other than techies) hear:

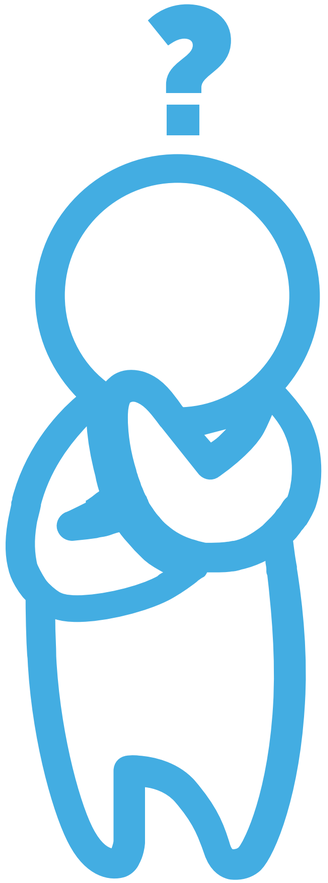
- “Nothing to do”
- “It comes for free, right?”
- “It’s not my concern - you’re the professional”

YET we need input as to the **degree** of quality that we need, or the risk-reward investment



Your role

- Identify key CFRs early
- Keep them alive in the minds of developers
- Balance “not enough” versus “too much” up front thinking
- Understand overall costs (now versus later)
- Approach it as risk management



EXERCISE

Going… Going… Gone!
Define the CFRs

- Choose 3 key CFRs
- Decide on measures
- Agree on how you meet them
- Can you automate them?

TECHNICAL VISION


Do Do Do Do Do
What's missing?

What's missing?

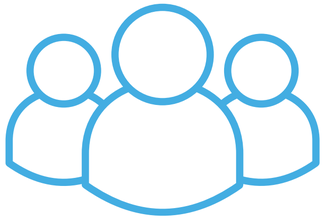
- What is our end goal?
- How are we going to get there?
- What's the next step?

Example

4 independent teams, each building microservices,
empowered to make any technicals.

Without any guidance,
what might we see after 2 years time?

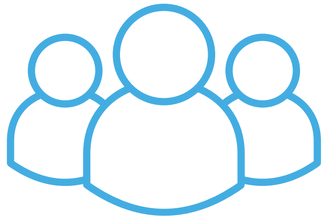
Example



React

Ruby on Rails

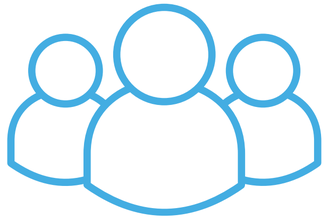
Selenium
Rspec



Vue

Spring

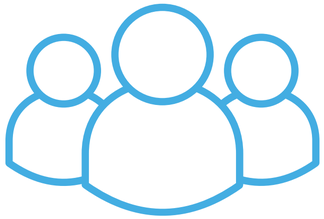
JUnit



Next.js

Micronaut

JBehave



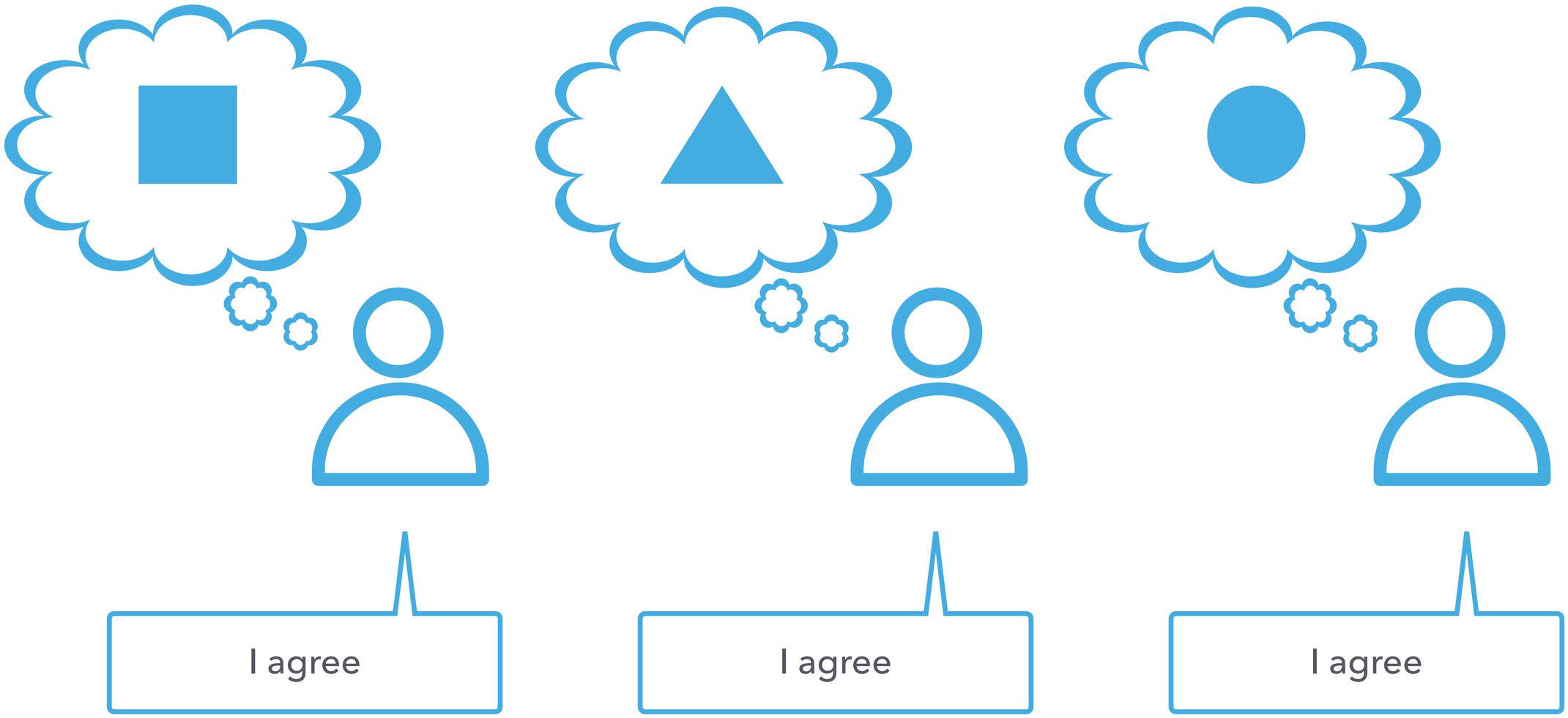
React

Ktor

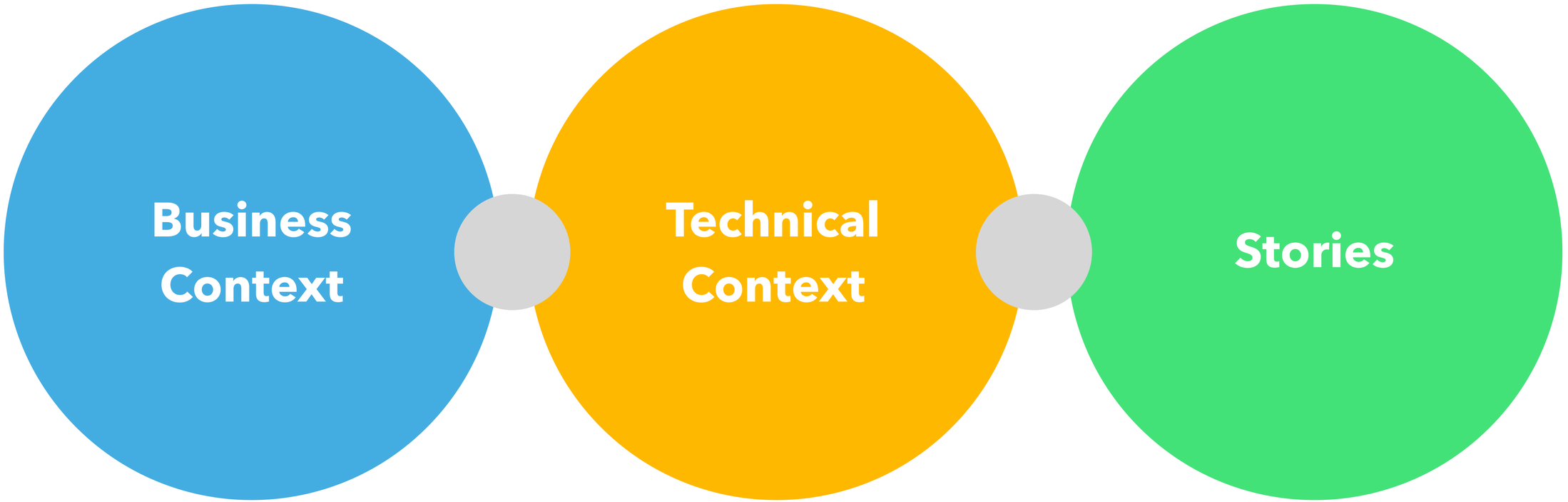
Spek

Note: The differences are not necessarily bad, but may be undesirable

Architecture is in our heads



What to cover



What to cover

- What's the business value?
- Who are the end users?
- What are other alternatives (perhaps offline processes)?

What to cover

- Outside in approach
- Follow a user journey through the system
- Avoid detail too early

- Use the C4 approach (at least the first two/three C's)
- Consider different perspectives

C4 MODEL

Context

Container diagram

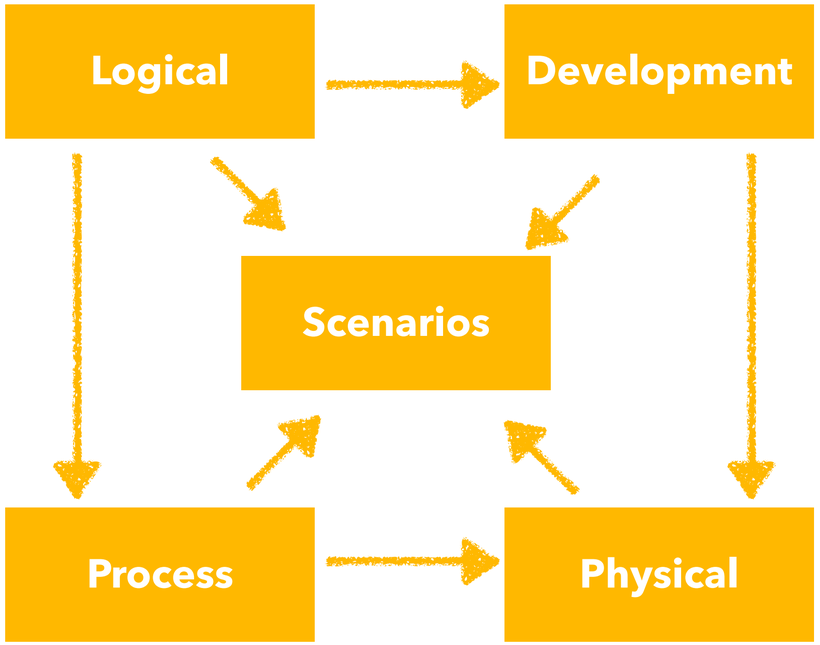
Component diagram

Class diagram

http://static.codingthearchitecture.com/c4.pdf

What to cover (later)

- 4+1 Krutchen view



What to cover

Invoke your inner shaman

- Use Architecture Decision Records (ADRs)
- Keep the vision alive
- Share lessons learned

Stories

FROM-TO EXAMPLE

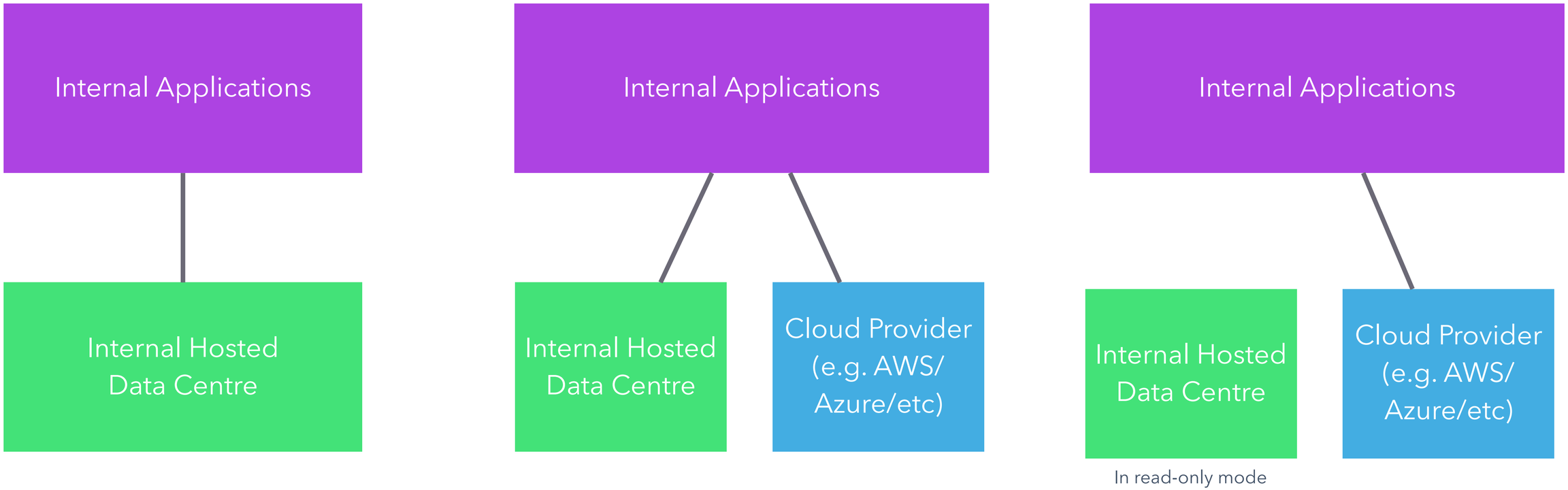
Example: Hosted to Cloud

Problem Statement:

Our business has 30 people building and maintaining our physical data centres. We spend approximately €12M each year on headcount, real estate, electricity costs.

Our development teams complain about long lead-times for new infrastructure and we physically cannot offer some infrastructure (e.g. data lake) with the same SLAs as cloud providers

Example: Hosted to Cloud



Consequences

End Goal

All services operating on a public cloud provider

How?

New services, built cloud-first

Old services, incrementally move to cloud

Next step?

For our services, we need to identify “seams” or “slices” and sequence to migrate

SYSTEM EXAMPLE

Example: Continuous Integration

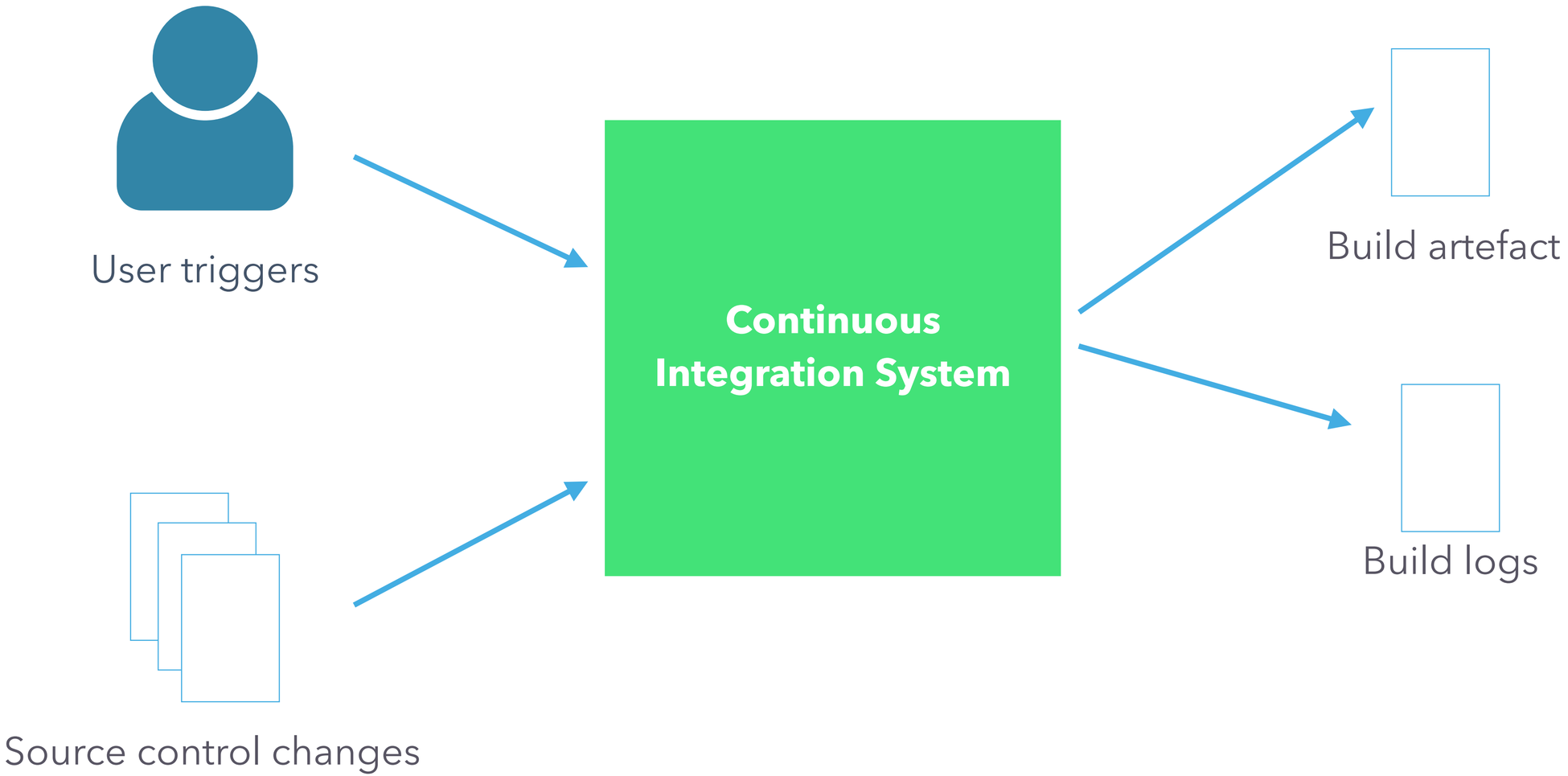
Problem Statement:

In the past, we used hire a “build team” or a “release manager”, whose role was to prepare a “releasable artefact”.

With agile practices, we believe in being in a state to be “releasable on demand.” Rather than have a team/person assemble the process, we want to automate the assembly of a “releasable artefact.”

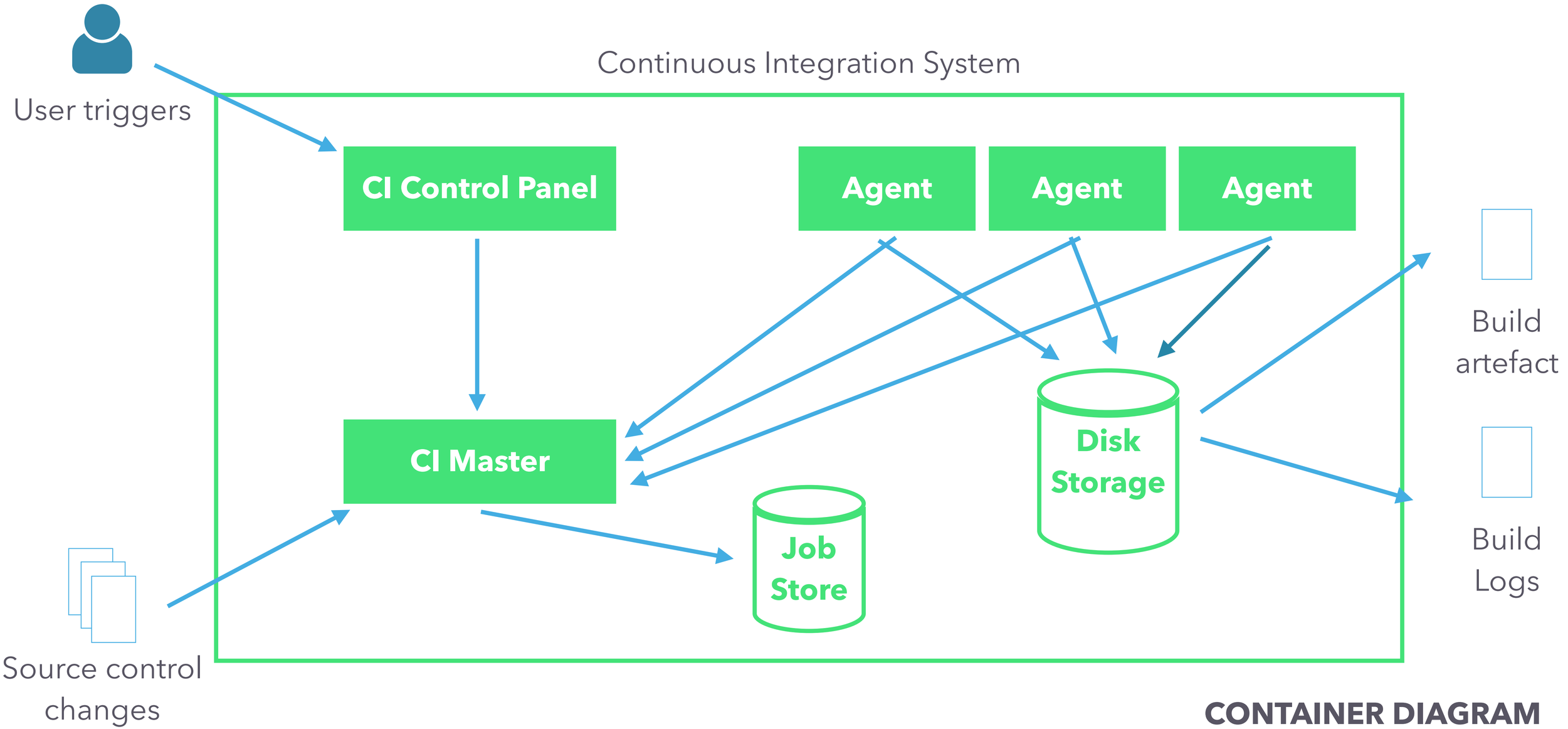
This process is called Continuous Integration.

Example: Continuous Integration



CONTEXT DIAGRAM

Example: Continuous Integration



CONTAINER DIAGRAM

Consequences

End Goal

A comprehensive Continuous Integration System

How?

Features per "Container"

Story planning around "Container Diagram"

Next step?

1. Build a MVP (tracer bullet)
2. Define Feature Leads for area
3. Prioritise features based on risk

EXERCISE

Single Page Architectural View

People

PEOPLE

Learning about your group strengths

*Pick **three words** from this list that best describe your personal strengths*
***Write one sticky note** for each word (three in total)*

Achiever
Activator
Adaptability
Analytical
Arranger
Belief
Command
Communication
Competition
Connectedness
Consistency
Context
Deliberative
Developer
Discipline
Empathy
Focus
Futuristic
Harmony
Ideation
Includer
Individualization
Input
Intellection
Learner
Maximizer
Positivity
Relator
Responsibility
Restorative
Self-Assurance
Significance
Strategic
Woo

Woo

Analytical

Achiever

Harmony

PEOPLE

Executing

Achiever
Arranger
Belief
Consistency
Discipline
Deliberative
Focus
Responsibility
Restorative

Achiever

Relationship Building

Adaptability
Connectedness
Developer
Empathy
Harmony
Includer
Individualization
Positivity
Relator

Harmony

Influencing

Activator
Command
Communication
Competition
Maximizer
Self-Assurance
Significance
Woo

Woo

Strategic Thinking

Analytical
Context
Futuristic
Ideation
Input
Intellection
Learner
Strategic

Analytical

Situational Leadership

EXERCISE PART 1

Discuss the distribution of strengths across your group.

Discuss: specifically

- If you have any strong clusters of strengths
- If you have strong individual strengths repeated
- What gaps do you have as a group?

Situational Leadership

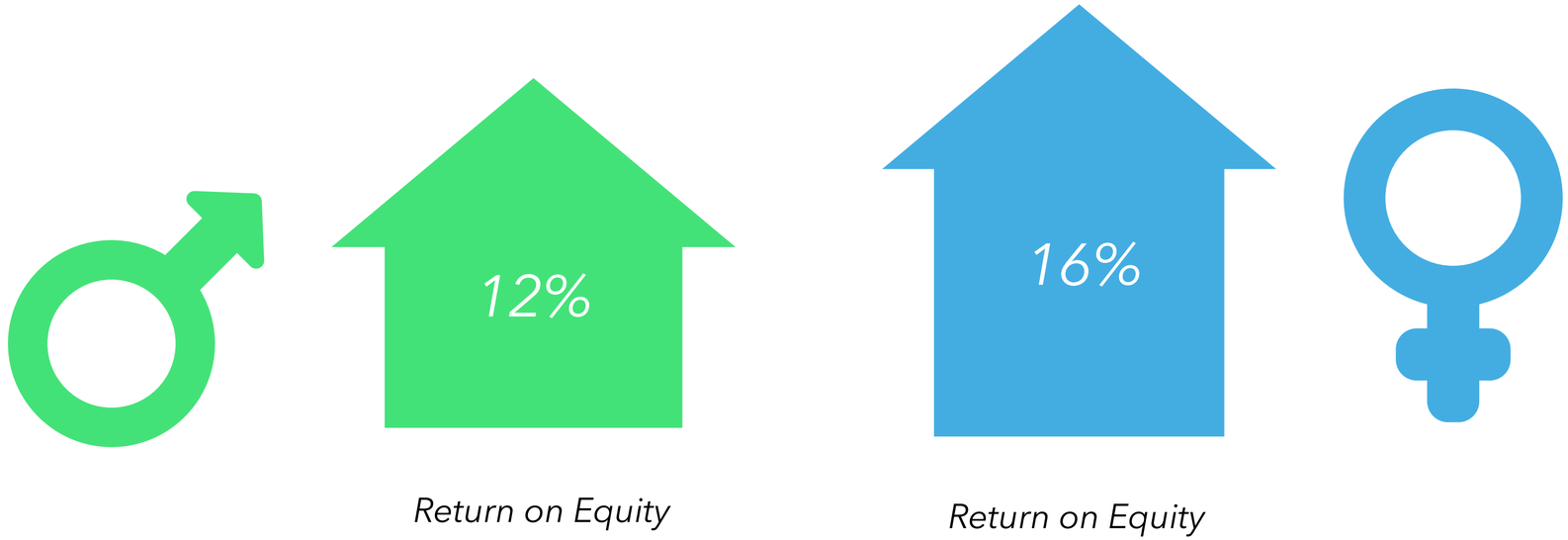
EXERCISE PART 2

Discuss the distribution of strengths across your group.

Come up with two scenarios:

1) A scenario where your group strengths would be **particularly useful**
2) A scenario where your lack of group strengths would be **challenging**

*"...over the past six years, companies with at least **some female board** representation **outperformed** those with no women on the board in terms of share price performance."*



Source: Credit Suisse Research Institute (August 2012) http://bit.ly/Oozuvl

PEOPLE

Mix that matters

DO	***DON'T***
Industry Background	*Academic background*
Country of origin	*Age*
Career path	
Gender	

PEOPLE

Source: *Mix that matters (Apriil 2017)* https://www.bcg.com/publications/2017/people-organization-leadership-talent-innovation-through-diversity-mix-that-matters

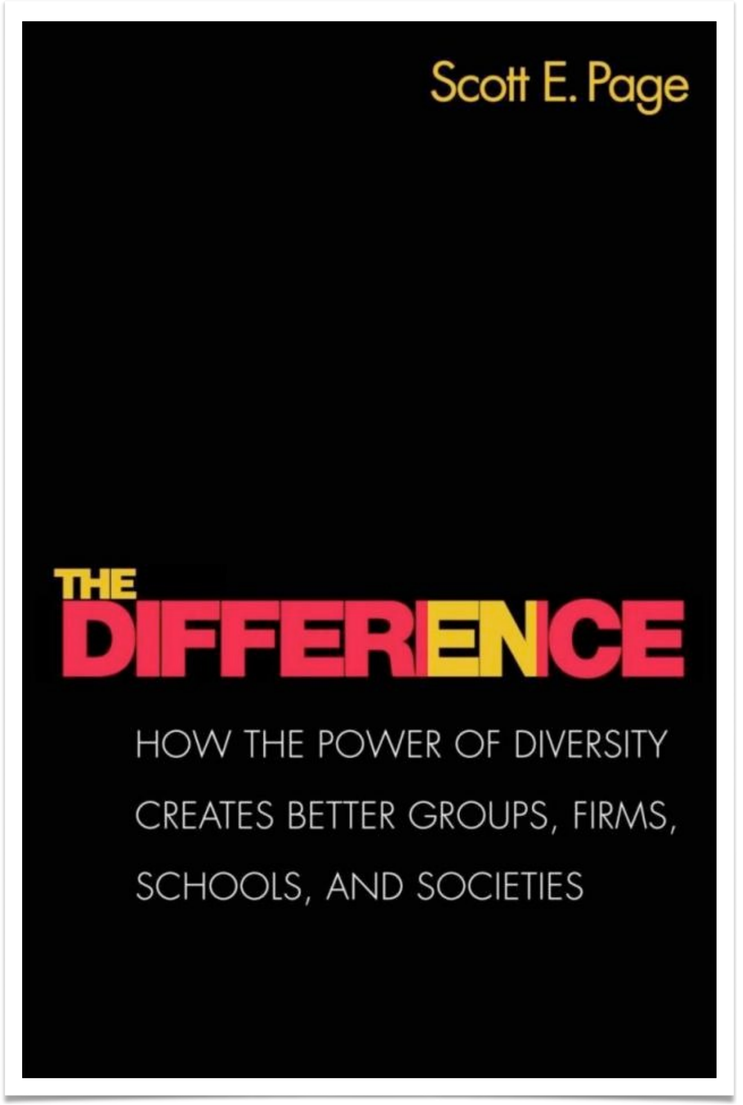
Collective Accuracy

=

Average Accuracy

*+ Diversity**

** Requires ability to integrate*



PEOPLE

Growing People

PEOPLE

PEOPLE

Source: Csikszentmihalyi, *Flow (1990)*

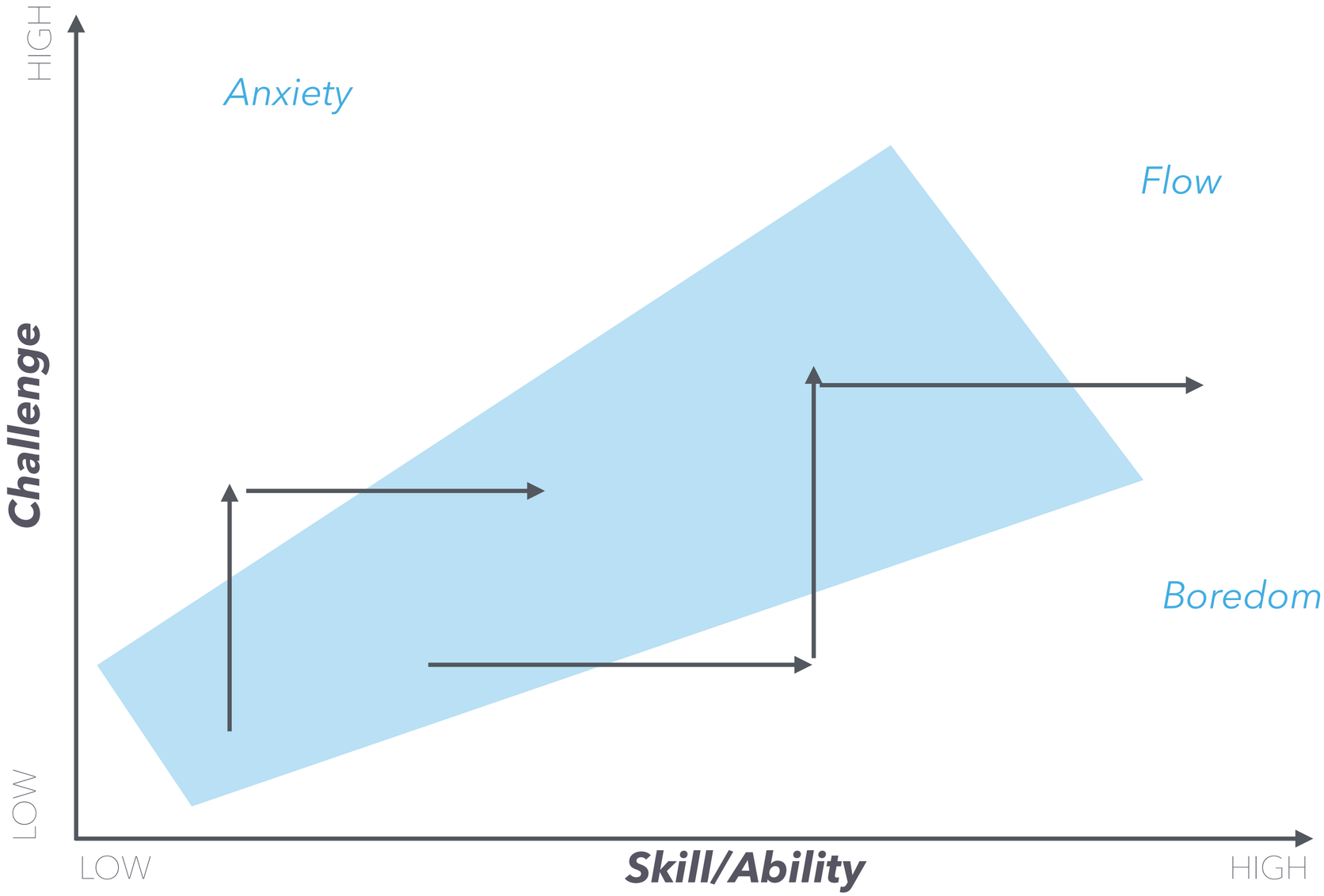
PEOPLE

Source: Csikszentmihalyi, Flow (1990)

PEOPLE

Source: Csikszentmihalyi, Flow (1990)

PEOPLE

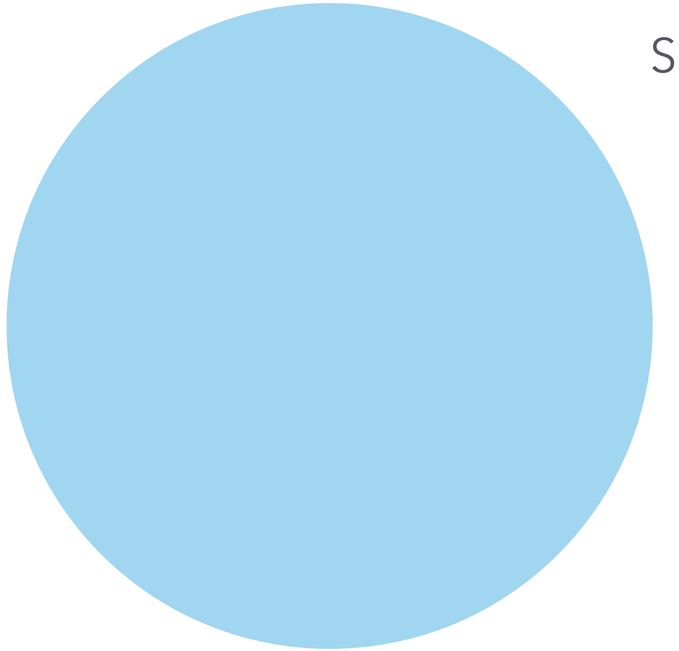


Source: Csikszentmihalyi, *Flow (1990)*

Maximising Potential

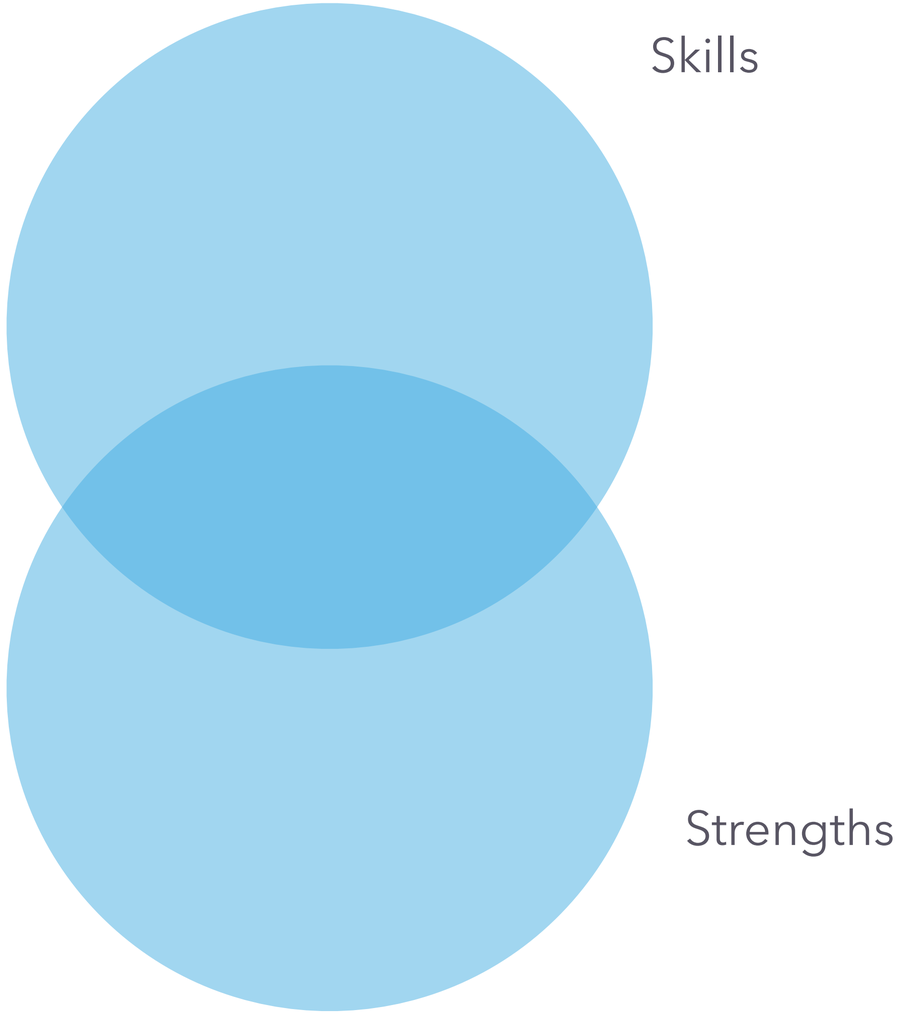
PEOPLE

PEOPLE

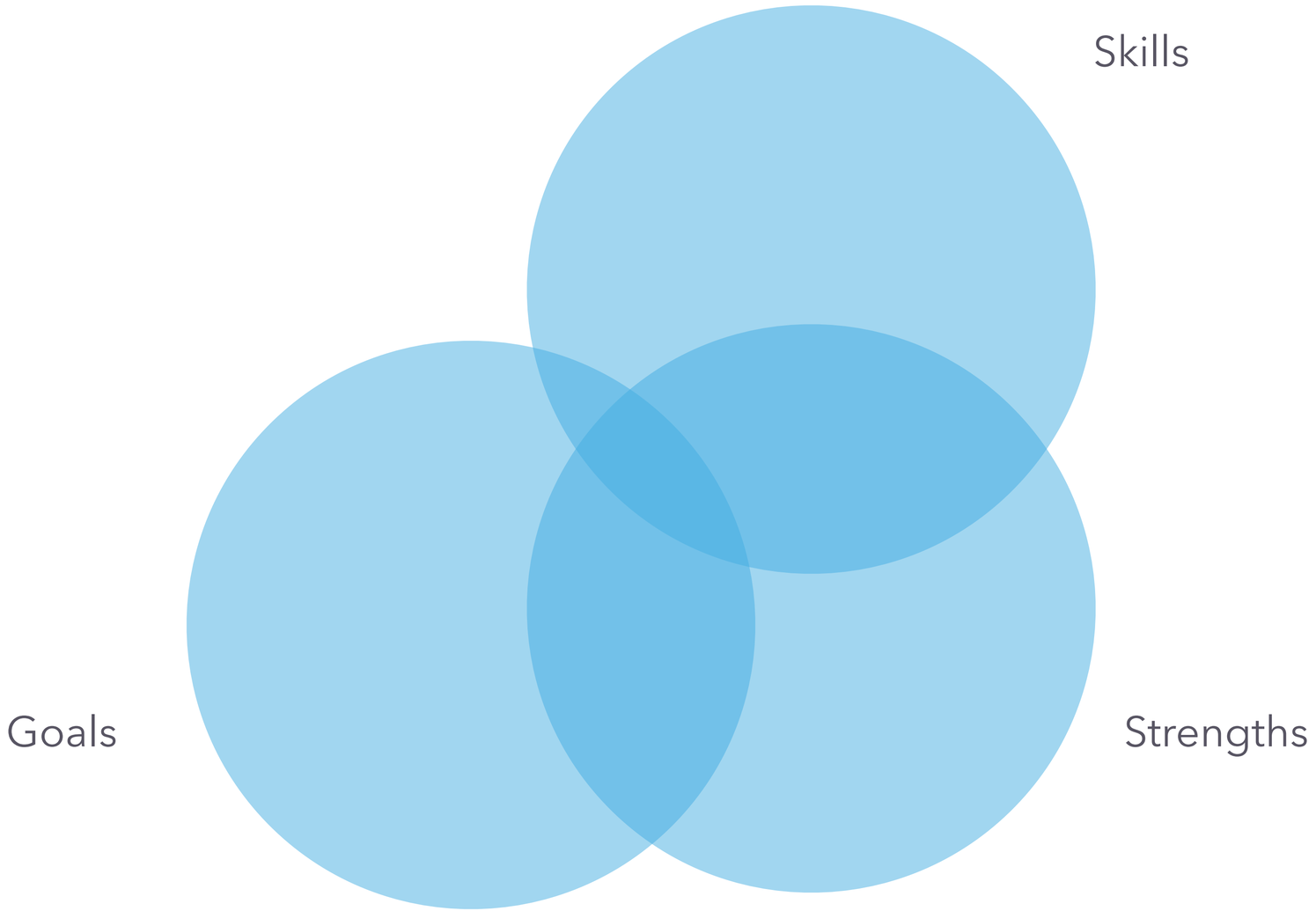


Skills

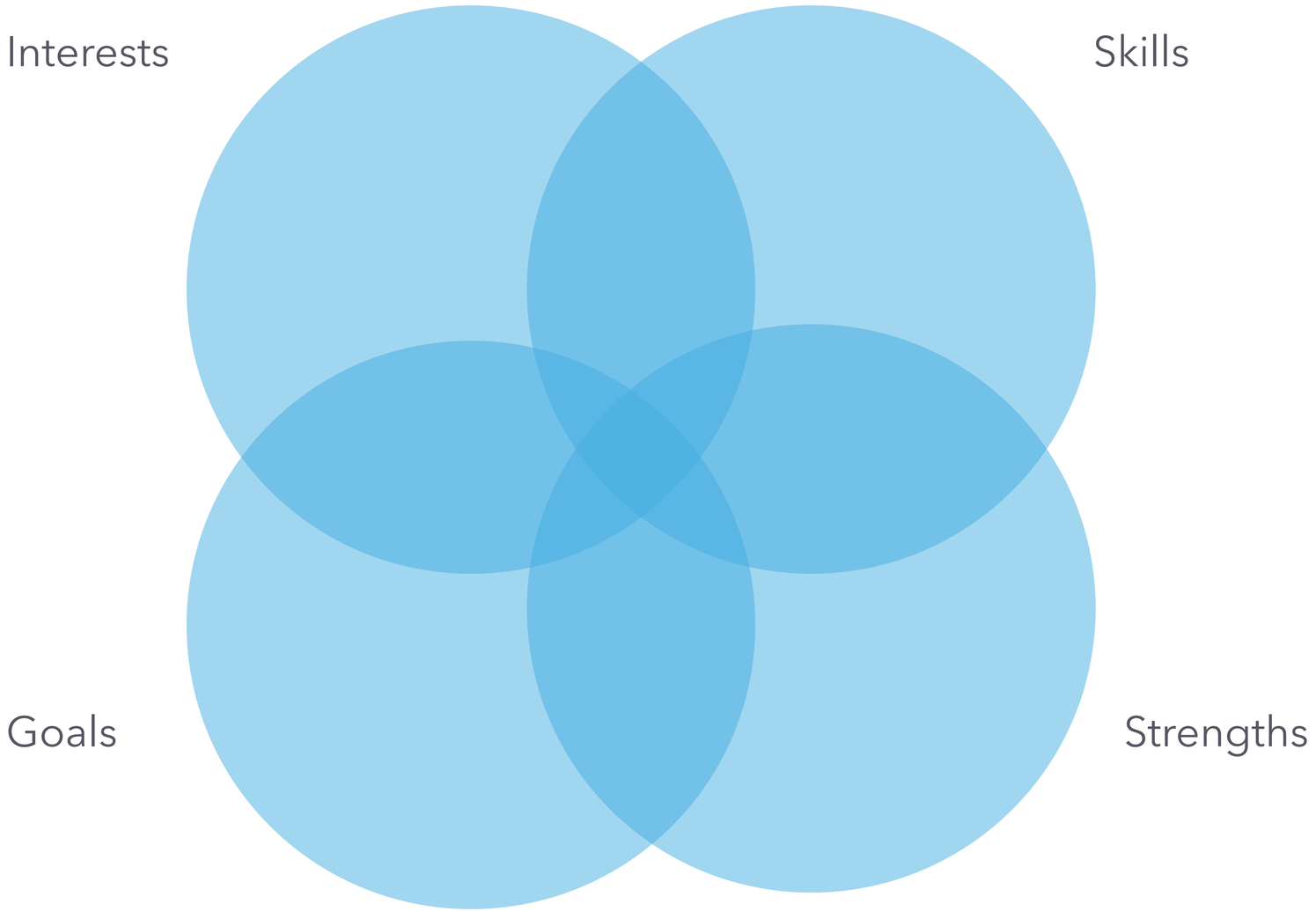
PEOPLE



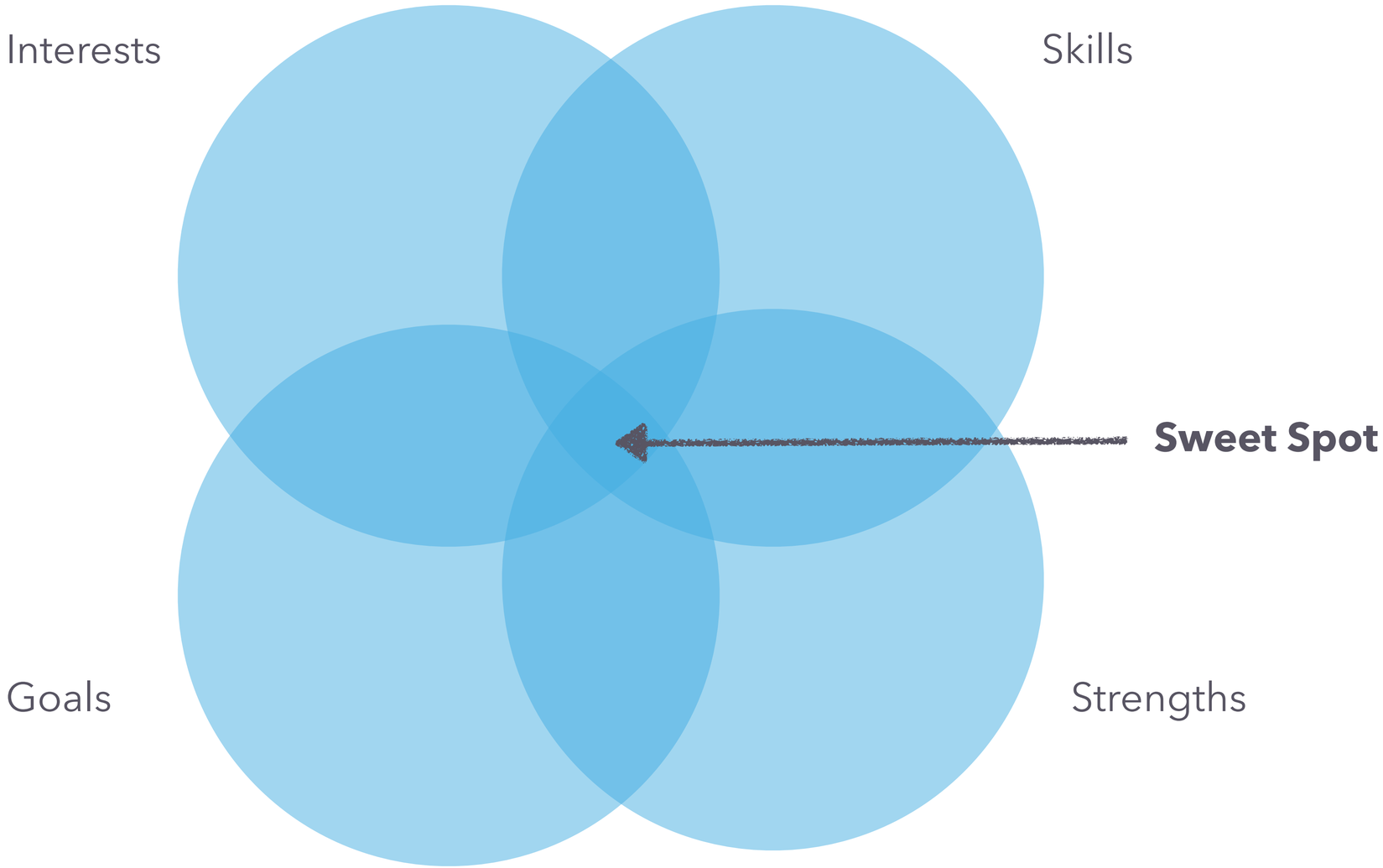
PEOPLE



PEOPLE



PEOPLE



Learning Activities

PEOPLE

Pair Programming

Team code reviews

Mob Programming

Learning Activities

Video or Book Club

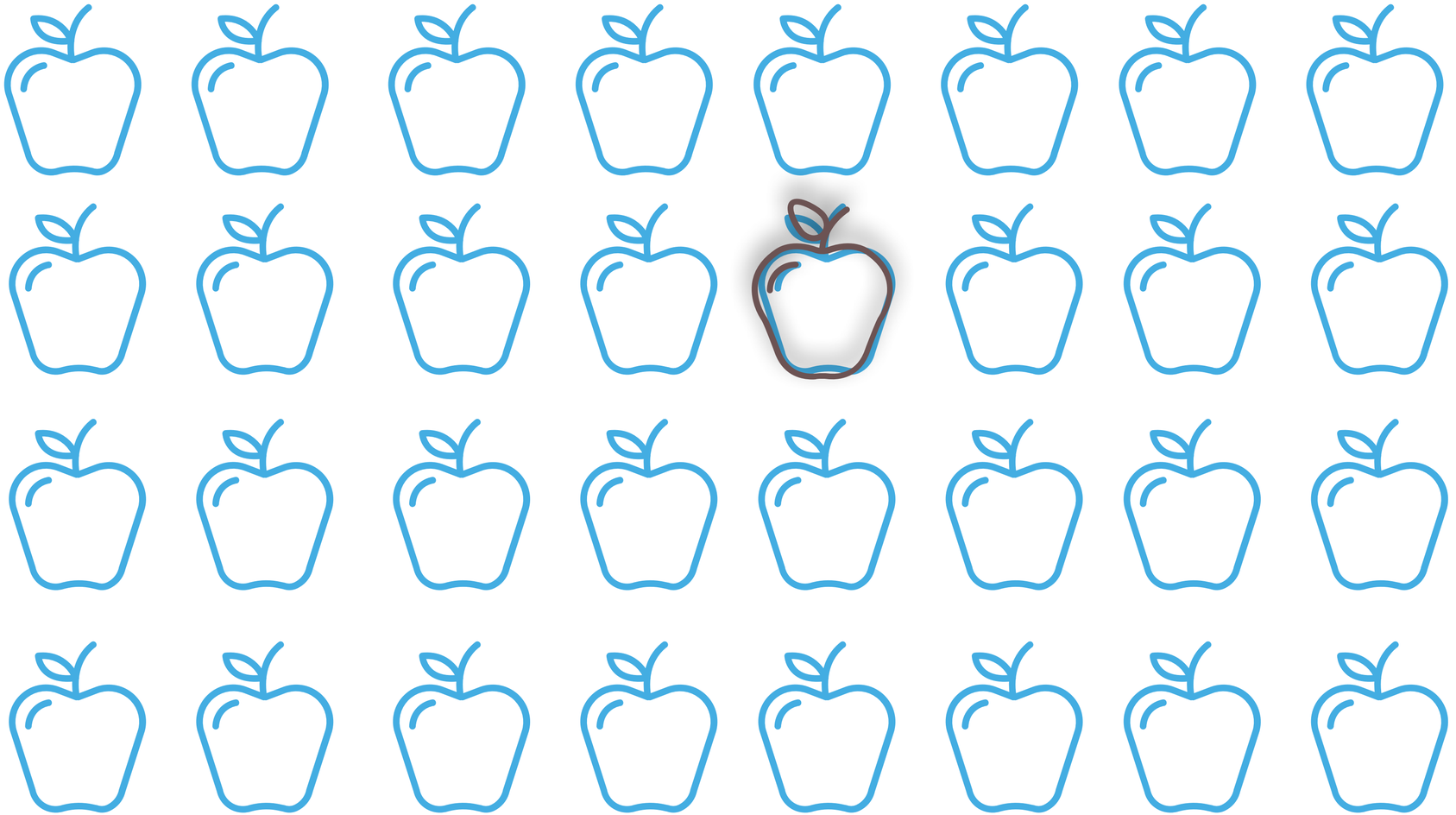
Brown Bag sessions

Technical Retrospectives

Spike Showcases

PEOPLE

Beware the Bad Apple



PEOPLE

Process

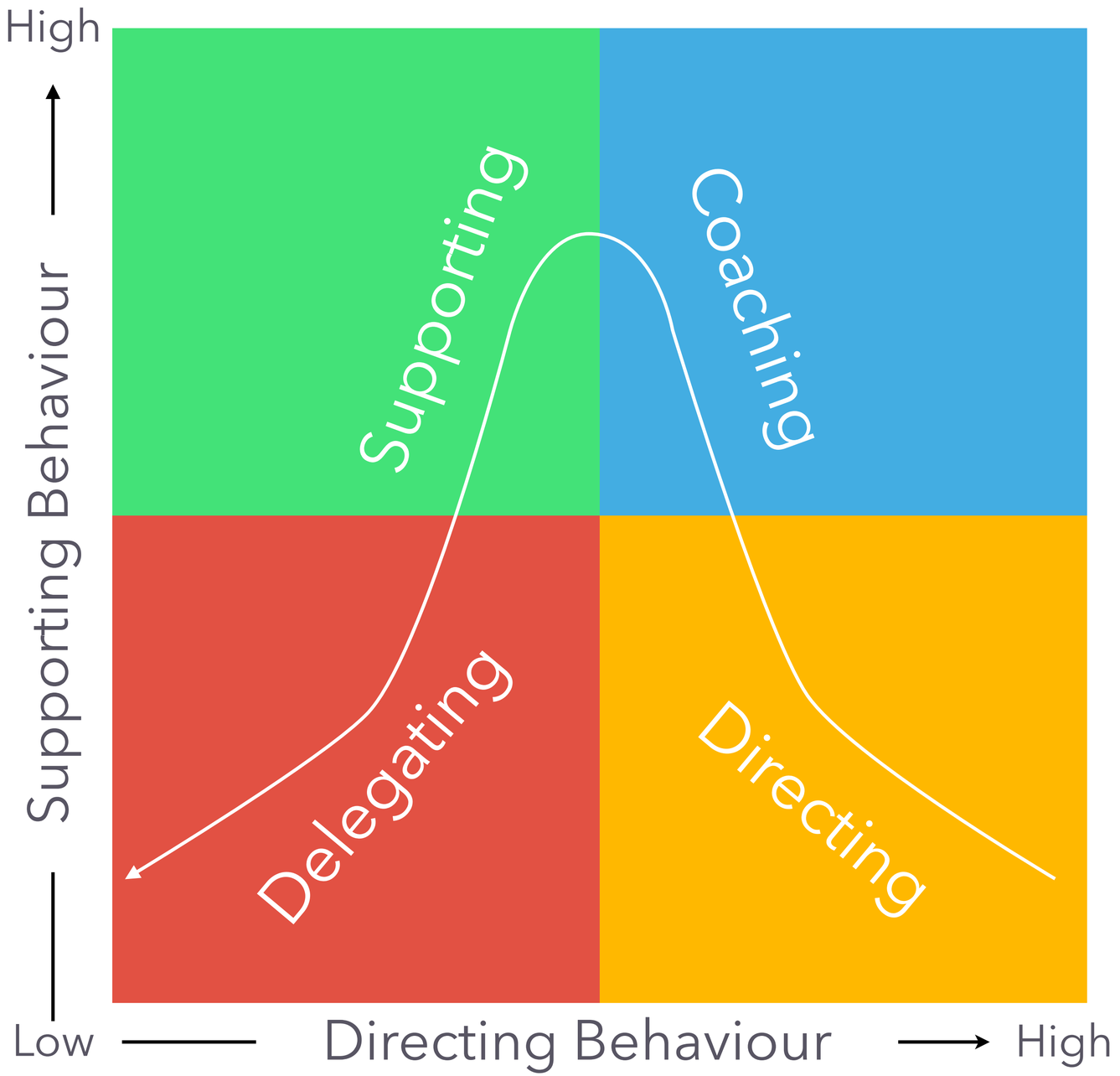
"Is it okay to tell people what to do?"

YES

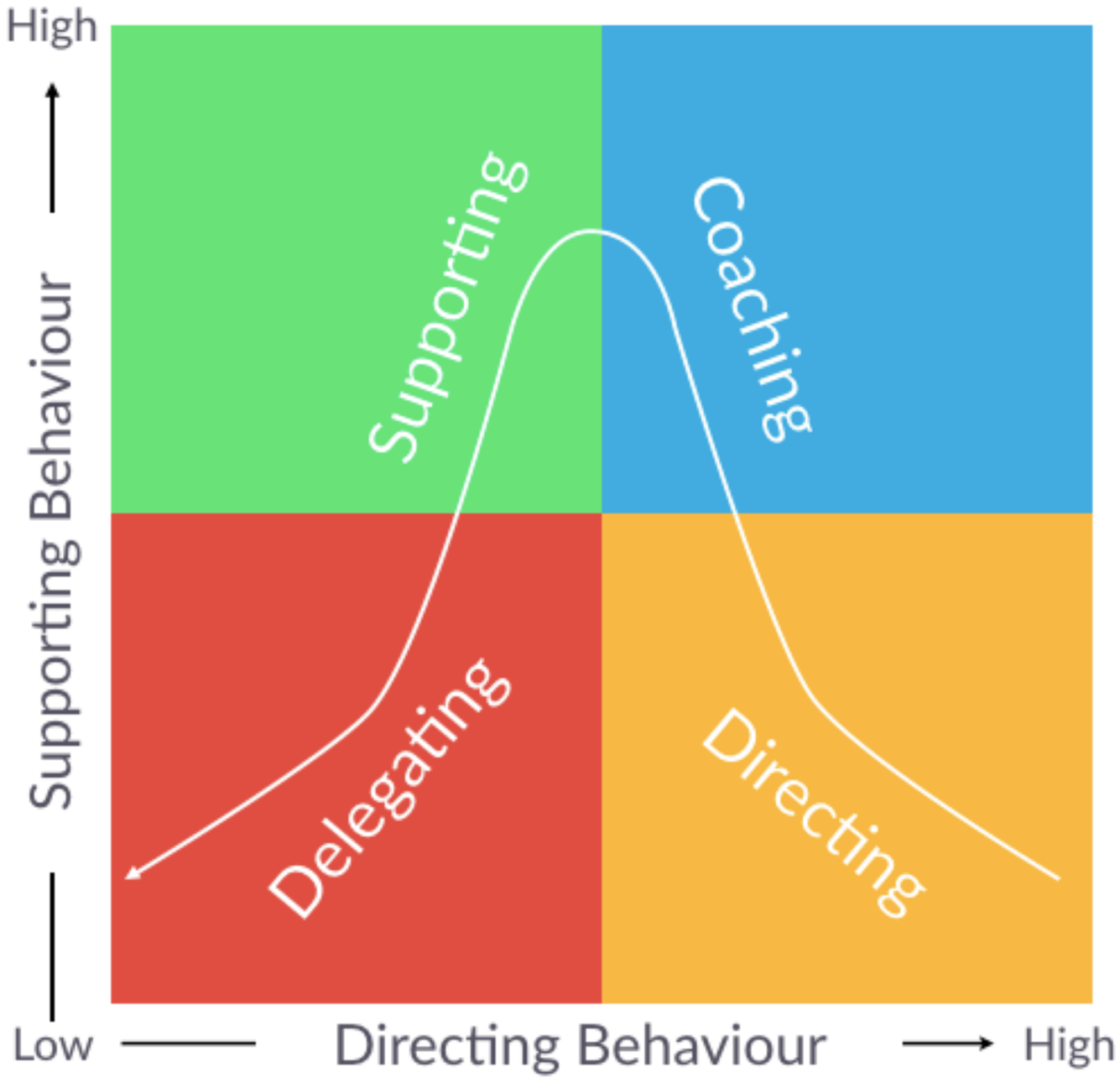
but only sometimes

PROCESS

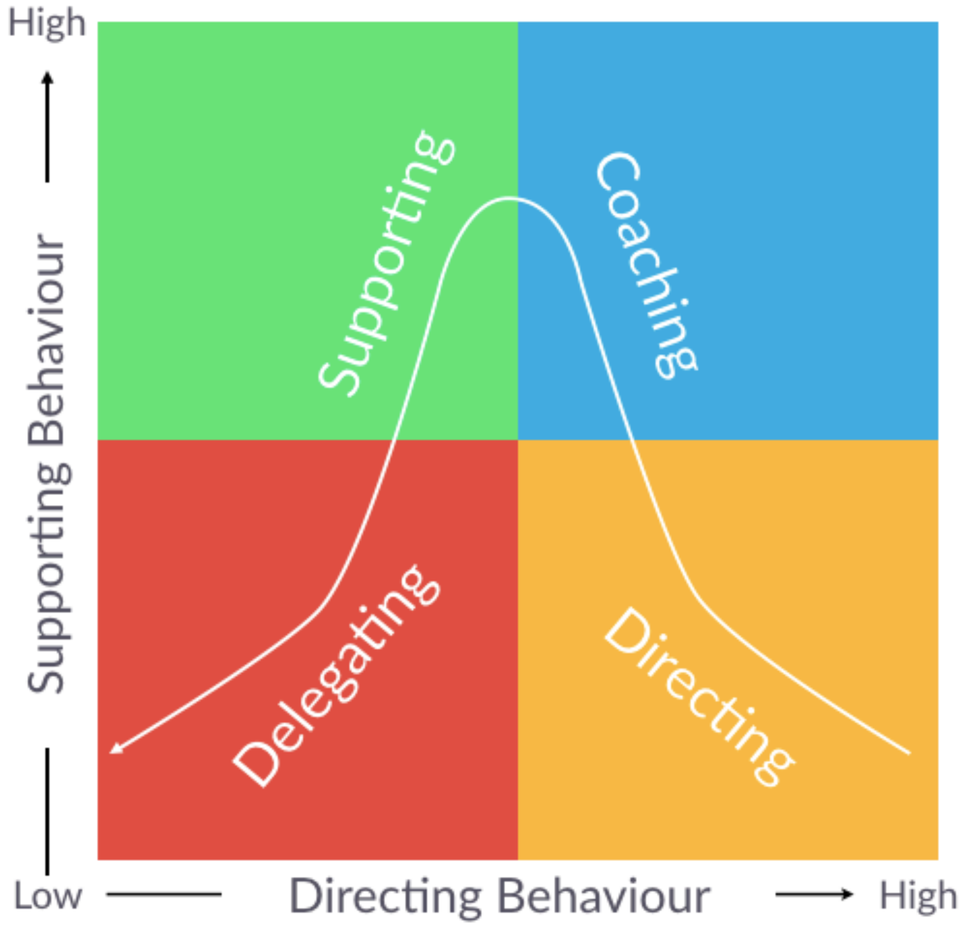
PROCESS

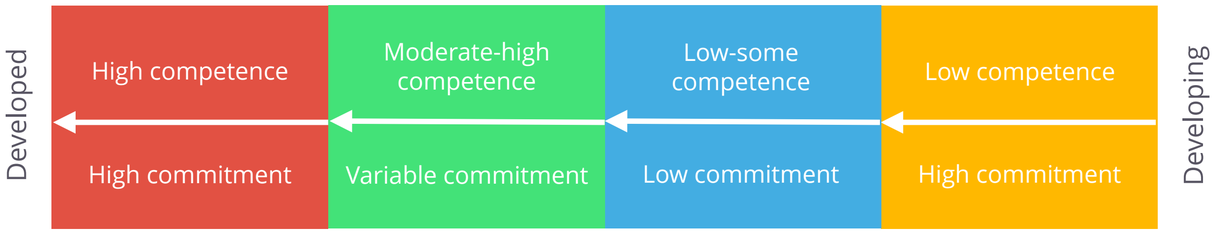

High
Supporting Behaviour
Low
Supporting
Coaching
Delegating
Directing
Directing Behaviour
High

PROCESS


High
Supporting Behaviour
Low
Supporting
Coaching
Delegating
Directing
Directing Behaviour
High

PROCESS

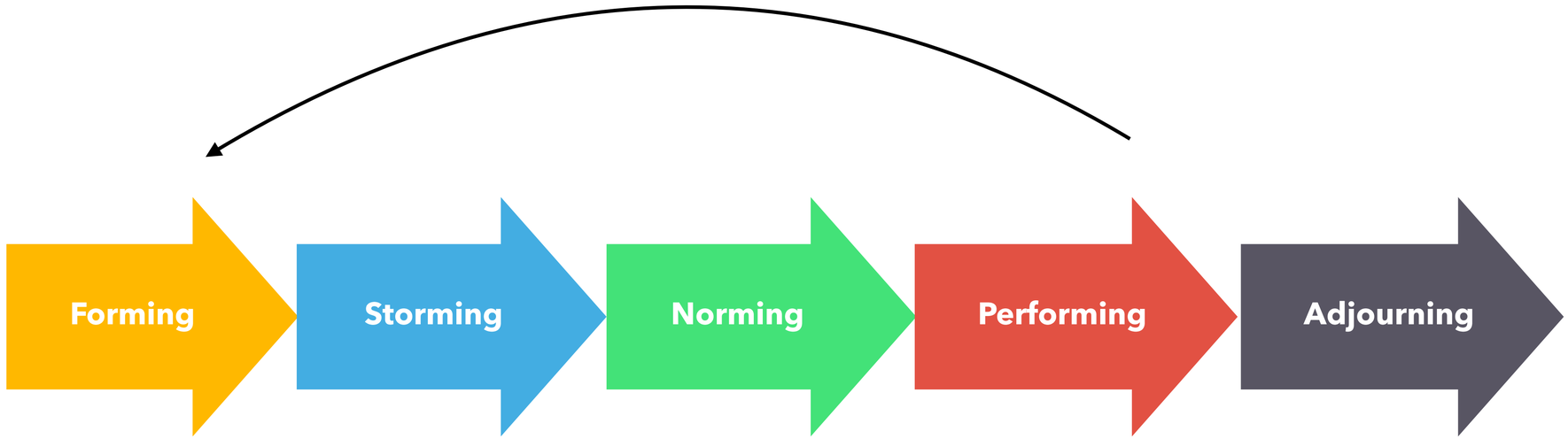




Development level of an individual

PROCESS

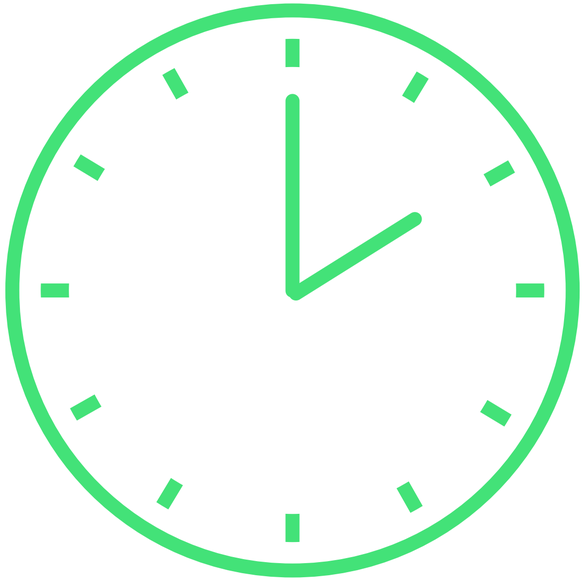
Tuckman's Model



Tuckman's Model

PROCESS

PROCESS

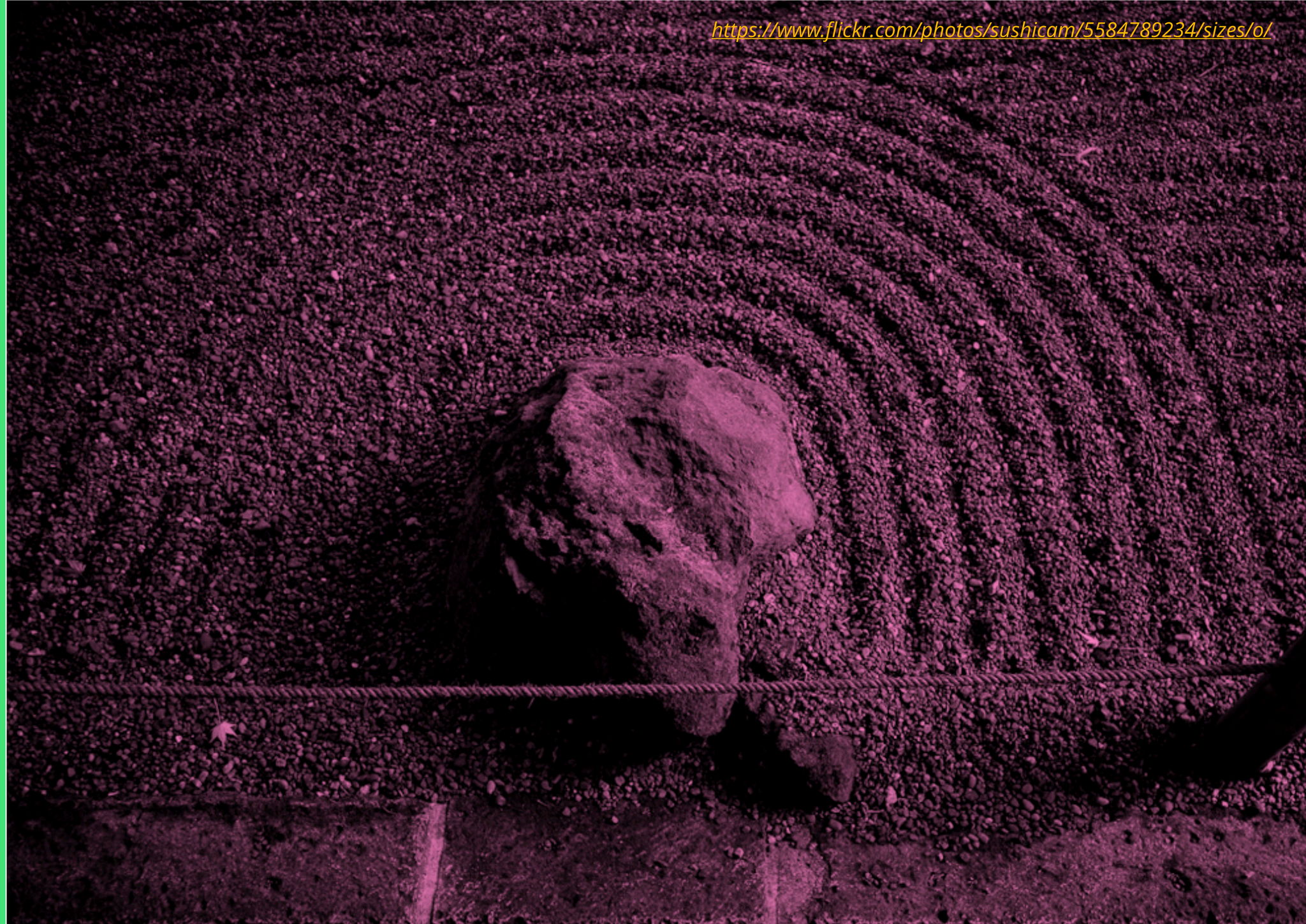


Make time for you

PROCESS

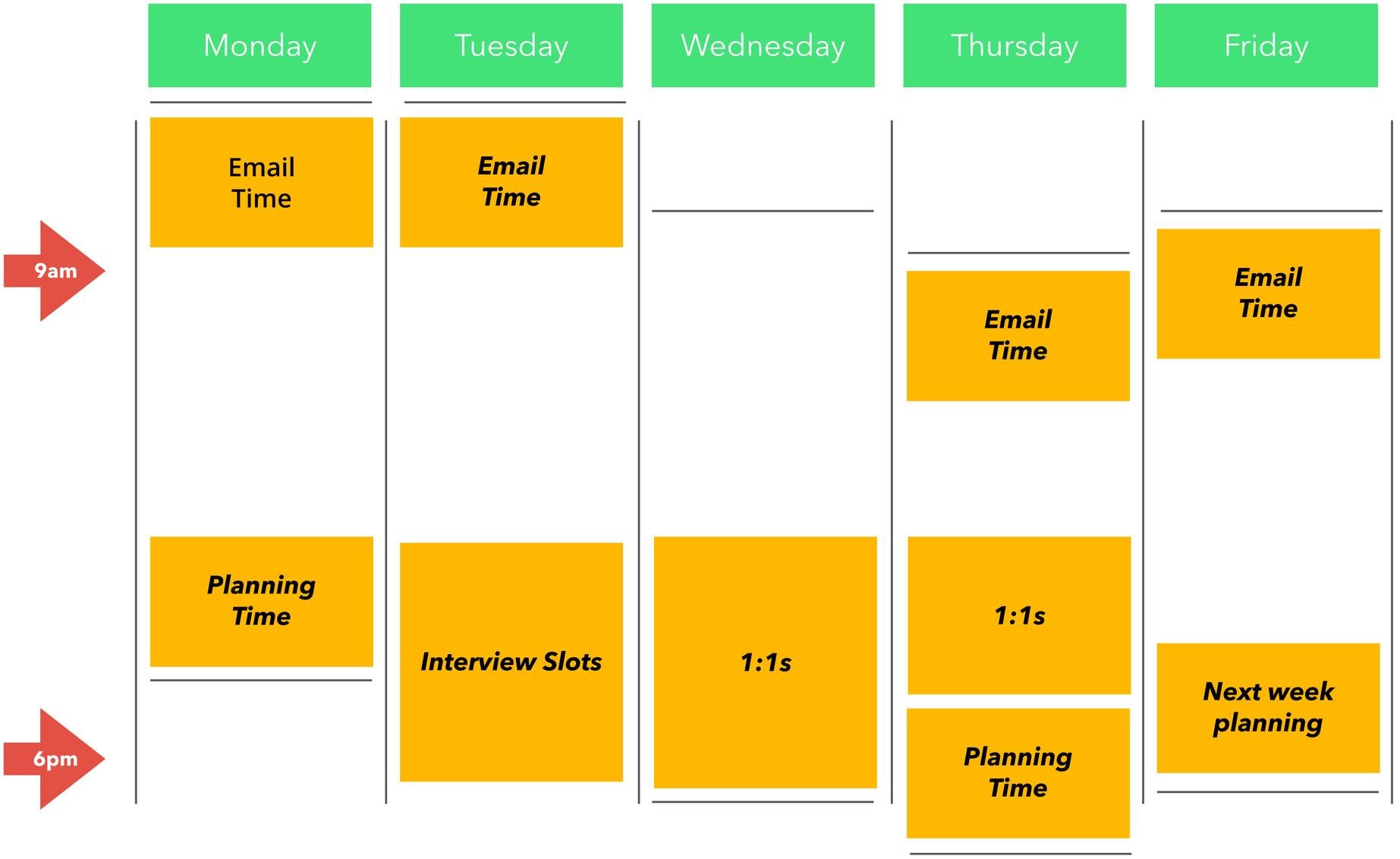
http://www.flickr.com/photos/mbeo52/5062096969/sizes/l/

PROCESS



https://www.flickr.com/photos/sushicam/5584789234/sizes/o/

PROCESS


Monday
Tuesday
Wednesday
Thursday
Friday
9am
6pm
Email Time
Planning Time
Email Time
Interview Slots
1:1s
Email Time
1:1s
Planning Time
Email Time
Next week planning

	Urgent	Not Urgent
Important	Do	Decide
Not Important	Delegate	Delete

PROCESS

Eisenhower Matrix

EXERCISE PART 1

As a group, brainstorm a list of 10 activities (tasks, meetings, etc) you have been doing the last 2 weeks.

Eisenhower Matrix

EXERCISE PART 2

Draw the matrix on a flip chart and then categorise your activities

"Essentially, all models are ***wrong****, but some are* ***useful****" - George E.P. Box*

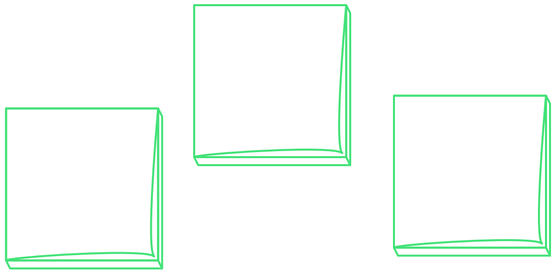
PROCESS

CONCLUDING THOUGHTS

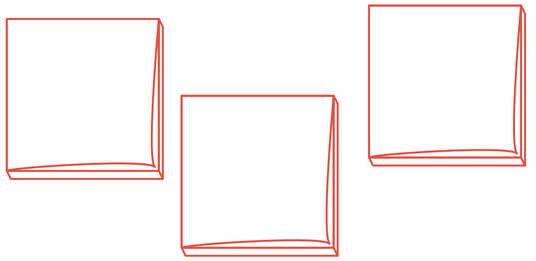
EXERCISE

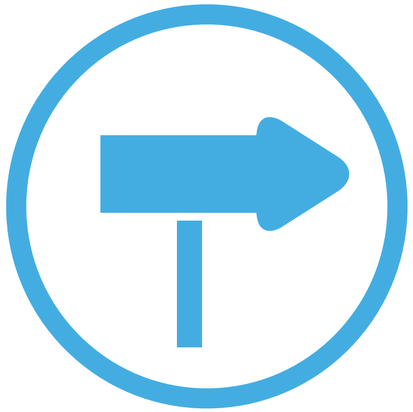
Hopes & Concerns Recap

What is your **biggest hope** today?



What **is your biggest concern** today (about the Tech Lead role)?





Action Plan

“Never stop learning because life never stops teaching”

Example Actions

Taking learning further

"Watch a conference talk about active listening in the next 3 weeks"

Example Actions

Taking learning further

"Pick one architecture book to read in the next 3 days"

Example Actions

Taking learning further

"Read (selected) book in the next three months"

Example Actions

Taking learning further

"Write a book summary and share with three colleagues/friends"

Example Actions

Taking learning further

"Meet 4 people for a coffee/tea I don't normally talk to in the next 3 weeks"

SMART Actions

Taking learning further

- **Specific** – Be concrete (“use a verb - e.g. do/read/write/talk”)
- **Measurable** – When do you know you’re done.
- **Attainable** – What results can realistically be achieved, given available resources and time?
- **Relevant** – Does this help you with your own goal?
- **Time-related** – When? Already given

Questions?