Techniques to become an effective agent of change

Successfully transitioning to a senior IC role

Jeremy Cook

Who?

Who?

Canadian with an English accent / Englishman with a Canadian passport

Who?

Canadian with an English accent / Englishman with a Canadian passport

Sometime musician, current software developer

Who?

Canadian with an English accent / Englishman with a Canadian passport

Sometime musician, current software developer

Senior principal developer at Oracle

What changes when you become a senior individual contributor?

What's the difference between a development manager and a senior IC?

One of them thinks they write code for a living…

You're a manager now, but you manage from the perspective of the code
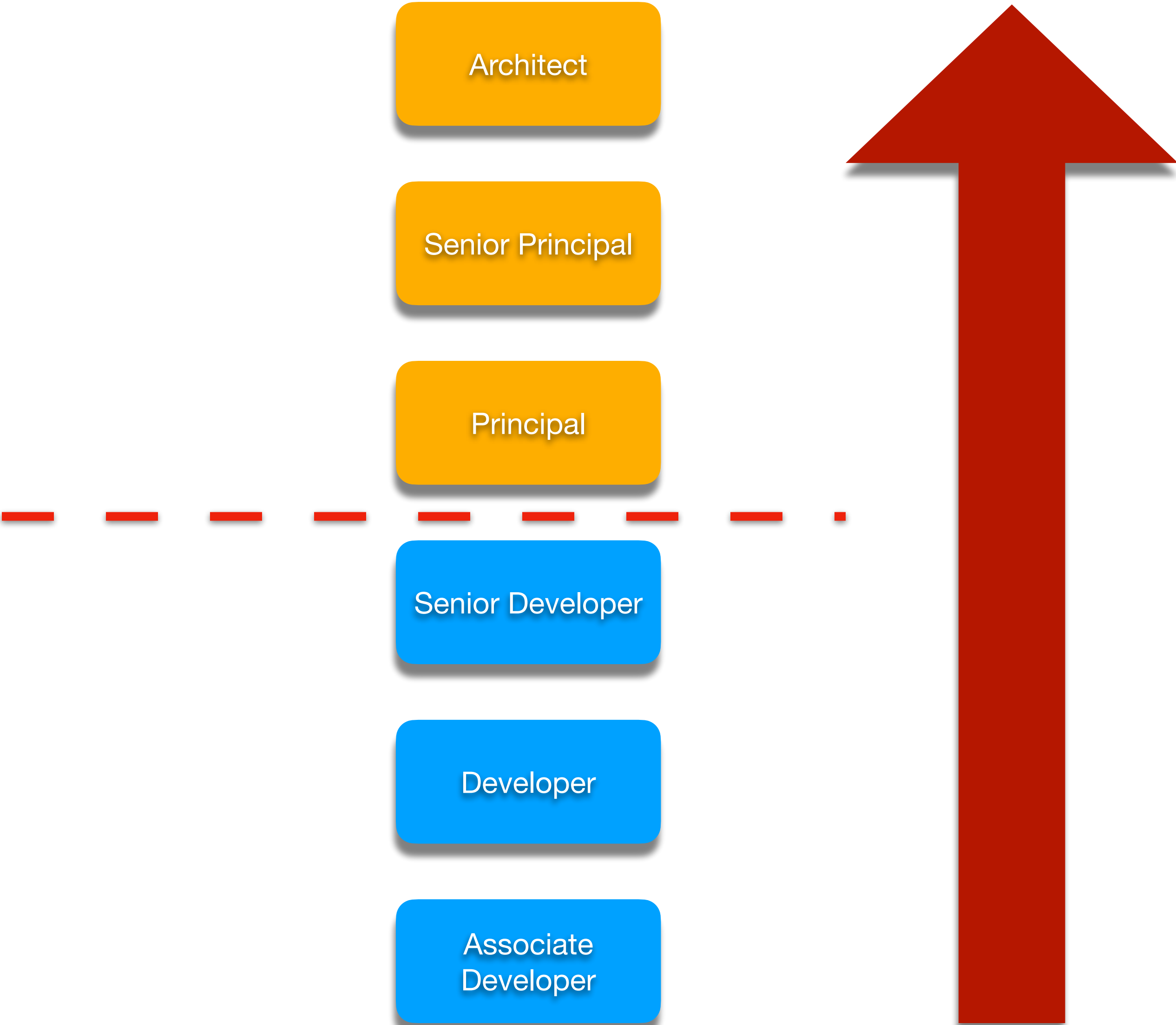
Career Progression as an Individual Contributor

Career Progression as an Individual Contributor
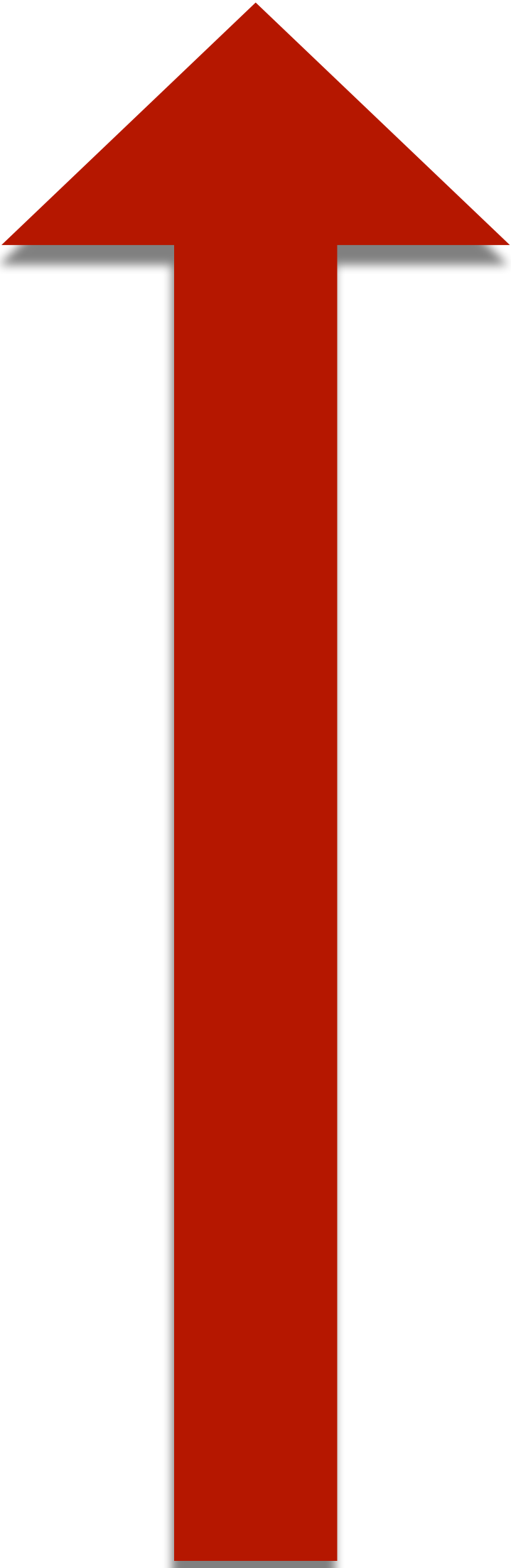
Architect

Senior Principal

Principal

Senior Developer

Developer

Associate Developer

Career Progression as an Individual Contributor

Architect

Senior Principal

Principal

Senior Developer

Developer

Associate Developer

Career Progression as an Individual Contributor

Architect

Senior Principal

Principal

Senior Developer

Developer

Associate Developer

Career Progression as an Individual Contributor

Architect

Senior Principal

Principal

Senior Developer

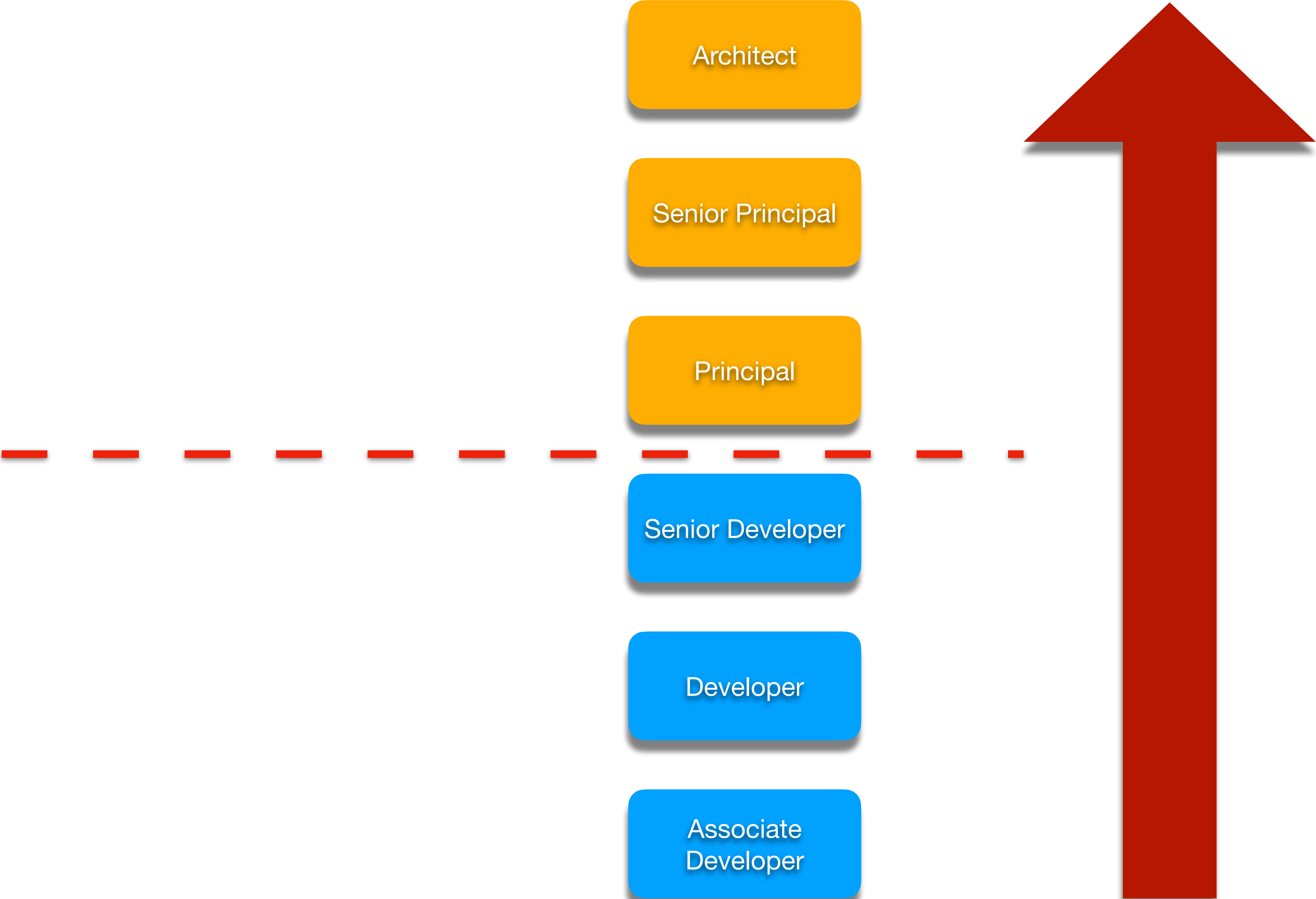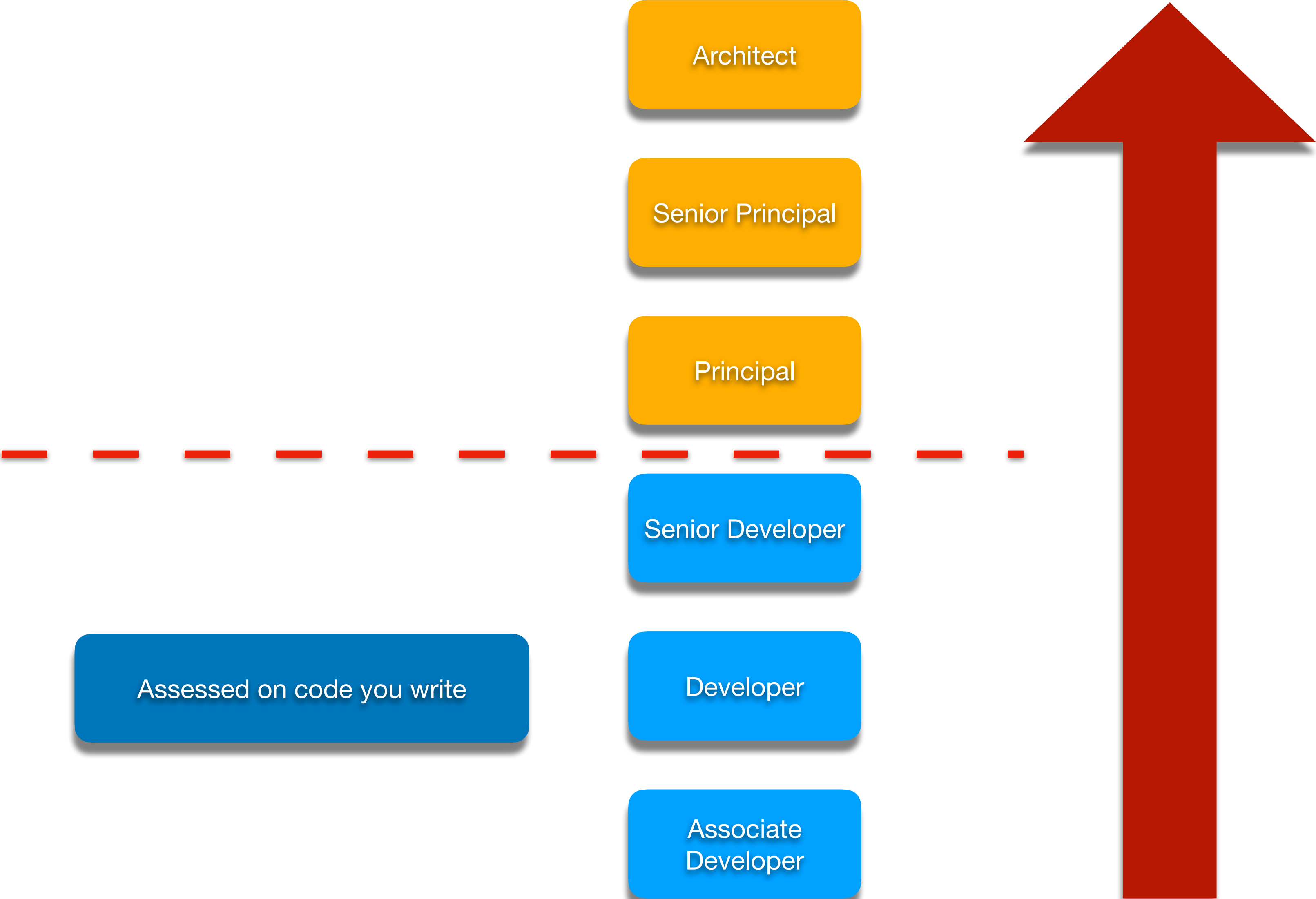Assessed on code you write

Developer

Associate Developer

Career Progression as an Individual Contributor

Architect

Senior Principal

Assessed on how you enable others

Principal

Senior Developer

Developer

Assessed on code you write

Associate Developer

Career Progression as an Individual Contributor

Architect

Senior Principal

Influence

Principal

Senior Developer

Impact

Developer

Associate Developer

Differences in expectations

Up to senior developer   **Principal developer and above**
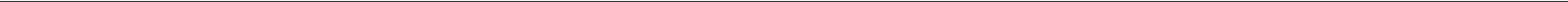
_____

Differences in expectations

Up to senior developer	Principal developer and above
Deliver code and features	

Differences in expectations

Up to senior developer	Principal developer and above
Deliver code and features	Enable others to deliver code and features

Differences in expectations

Up to senior developer	Principal developer and above
Deliver code and features	Enable others to deliver code and features
Work in an engineering direction defined by others	

Differences in expectations

Up to senior developer	Principal developer and above
Deliver code and features	Enable others to deliver code and features
Work in an engineering direction defined by others	Influence and define engineering direction

Differences in expectations

Up to senior developer	Principal developer and above
Deliver code and features	Enable others to deliver code and features
Work in an engineering direction defined by others	Influence and define engineering direction
Short term feature driven work	

Differences in expectations

Up to senior developer	Principal developer and above
Deliver code and features	Enable others to deliver code and features
Work in an engineering direction defined by others	Influence and define engineering direction
Short term feature driven work	Long term, often strategic, work

Differences in expectations

Up to senior developer	Principal developer and above
Deliver code and features	Enable others to deliver code and features
Work in an engineering direction defined by others	Influence and define engineering direction
Short term feature driven work	Long term, often strategic, work
Spend time implementing	

Differences in expectations

Up to senior developer	Principal developer and above
Deliver code and features	Enable others to deliver code and features
Work in an engineering direction defined by others	Influence and define engineering direction
Short term feature driven work	Long term, often strategic, work
Spend time implementing	Spend time exploring

You are an engineering leader…

...but you lead without authority

Everything is a people problem

Everything is a people problem

Everything is a people problem

Code is easy, **People are hard**

Everything is a people problem

Code is easy, **People are hard**

Your **job is convincing people**

Disclaimer!

Driving technical change

How do you make technical changes 'stick'?

Driving technical change

Driving technical change

Many **change management techniques**, e.g. PDCA, ADKAR, Kotter

Driving technical change

Many **change management techniques**, e.g. PDCA, ADKAR, Kotter

Address **people side of changing** an organization

Driving technical change

Many **change management techniques**, e.g. PDCA, ADKAR, Kotter

Address **people side of changing** an organization

Four underlying principles support technical change
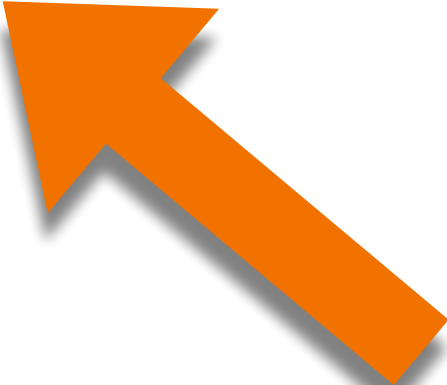
Mandate

Measurement

Shared Ownership

Enablement

Mandate

Mandate

Who has **authority to declare the change** is happening?

Mandate

Who has **authority to declare the change** is happening?

Bigger change ➡ Greater authority

Mandate

Who has **authority to declare the change** is happening?

Bigger change ➡ Greater authority

Identify **allies,** build **consensus**

Measurement

Measurement

"You can't optimize what you can't measure"

Measurement

"You can't optimize what you can't measure"

Make **progress apparent** to everyone

Measurement

"You can't optimize what you can't measure"

Make **progress apparent** to everyone

Decide how change measured and tell everyone

Shared Ownership

Shared Ownership

Get everyone invested

Shared Ownership

Get everyone invested

Create expectations and **delegate to others**

Shared Ownership

Get everyone invested

Create expectations and **delegate to others**

Celebrate success, understand failure to do better

Enablement

Enablement

Ensuring everyone has skills and tools needed to succeed

Enablement

Ensuring everyone has skills and tools needed to succeed

Pact between management/senior IC's and teams

Enablement

Ensuring everyone has skills and tools needed to succeed

Pact between management/senior IC's and teams

Continuous feedback to identify enablement needs

Techniques and tips

Be authentic

Be authentic

Be authentic

How can you expect others to adopt ideas you don't follow?

Be authentic

How can you expect others to adopt ideas you don't follow?

You **don't need** to be an **expert**

Be authentic

How can you expect others to adopt ideas you don't follow?

You **don't need** to be an **expert**

Need to **be an example** to others

Seek first to understand, then be understood

"Most people do not listen with the intent to understand; they listen with the intent to reply."

Dr. Stephen R. Covey

Seek first to understand, then be understood

Seek first to understand, then be understood

Understanding others ➡ greater influence

Seek first to understand, then be understood

Understanding others ➡ greater influence

Empathic listening: Understand, Reflect, Clarifying Questions

Seek first to understand, then be understood

Understanding others ➡ **greater influence**

Empathic listening: Understand, Reflect, Clarifying Questions

Respectfully state your **point of view**

Seek first to understand, then be understood

Understanding others ➡ **greater influence**

Empathic listening: Understand, Reflect, Clarifying Questions

Respectfully state your **point of view**

Makes you **more open to new insights**

Create time to practice

Skill = Knowledge + 10,000 times

Skill = Knowledge + Practice

Create time to practice

Create time to practice

All **professionals practice**

Create time to practice

All **professionals practice**

Knowledge != Skill

Create time to practice

All **professionals practice**

Knowledge != Skill

Learning **doesn't happen while executing**

Don't practice until you get it right.
Practice until you can't get it wrong.

Create time to practice

Create time to practice

Provide **time to dedicate to learning and practice**

Create time to practice

Provide **time to dedicate to learning and practice**

Encourage **team based formats**

Create time to practice

Provide **time to dedicate to learning and practice**

Encourage **team based formats**

Coach where needed

Create time to practice

Provide **time to dedicate to learning and practice**

Encourage **team based formats**

Coach where needed

Align with **engineering goals**

Spend one on one time with developers and teams

Spend one on one time with developers and teams

Spend one on one time with developers and teams

Incredibly important technique: make time for it

Spend one on one time with developers and teams

Incredibly important technique: make time for it

Identify key individuals/teams to spend time with

Spend one on one time with developers and teams

Incredibly important technique: make time for it

Identify key individuals/teams to spend time with

Reach out and book time

Demonstration defeats discussion

Demonstration defeats discussion

Demonstration defeats discussion

Sometimes it's **quicker to just demonstrate**…

Demonstration defeats discussion

Sometimes it's **quicker to just demonstrate**…
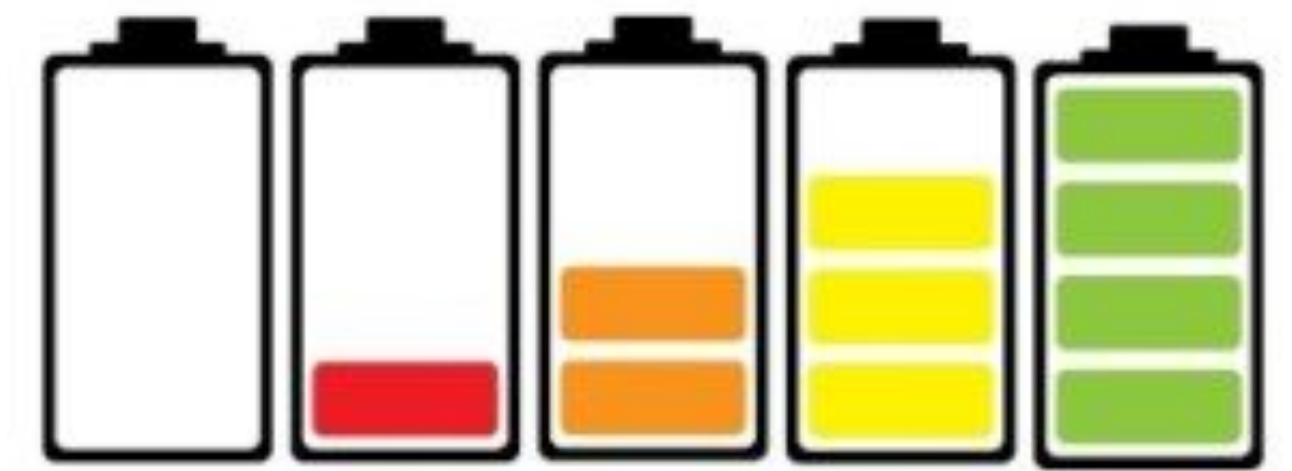
People approach new ideas with scepticism

Demonstration defeats discussion

Sometimes it's **quicker to just demonstrate**…

People approach new ideas with scepticism

If discussion is taking longer than just doing it, demonstrate

Pay attention to your trust battery

"Another concept we talk a lot about is something called a 'trust battery'…. every time you work with someone… the trust battery between the two of you is either charged or discharged."

Tobias Lütke

Pay attention to your trust battery

Pay attention to your trust battery

Need to be trusted to succeed

Pay attention to your trust battery

Need to be trusted to succeed

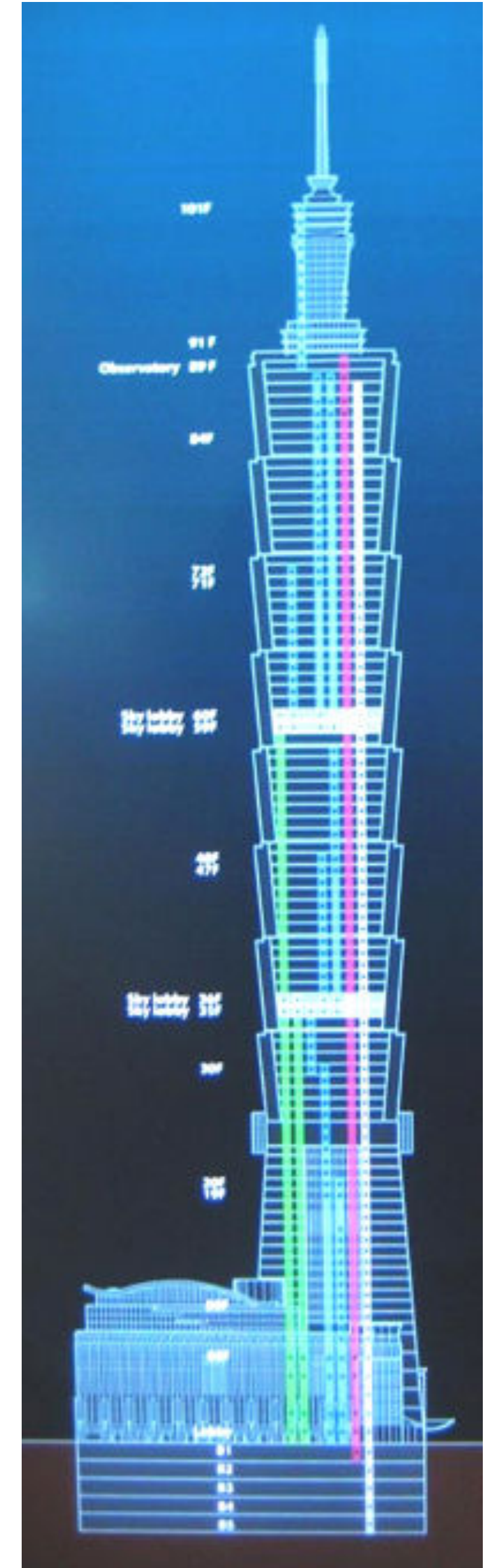**Many** 'trust batteries' **to manage**

Pay attention to your trust battery

Need to be trusted to succeed

Many 'trust batteries' **to manage**

Trust batteries **not always equal**…

Ride the organizational elevator

Ride the organizational elevator

Ride the organizational elevator

Need to **communicate effectively** with many levels

Ride the organizational elevator

Need to **communicate effectively** with many levels

Different people need **different information**

Ride the organizational elevator

Need to **communicate effectively** with many levels

Different people need **different information**

"**What does the person I'm talking to need to know to support me**?"

Maintain technical depth while building breadth

Maintain technical depth while building breadth

Maintain technical depth while building breadth

Depth -> **things you know**

Maintain technical depth while building breadth

Depth -> **things you know**

Breadth -> **things you know you don't know**

Maintain technical depth while building breadth

Depth -> **things you know**

Breadth -> **things you know you don't know**

Cultivating breadth allows you to offer more suggestions

Maintain technical depth while building breadth

Depth -> **things you know**

Breadth -> **things you know you don't know**

Cultivating breadth allows you to offer more suggestions

Beware of moving too much into things you know!

You will occasionally fail…

You will occasionally fail…

You will occasionally fail…

Change can involve experimentation

You will occasionally fail…

Change can involve experimentation

When **unsure of the outcome**, make it **clear in advance**

You will occasionally fail…

Change can involve experimentation

When **unsure of the outcome**, make it **clear in advance**

Continuously **learn to adapt**

You will occasionally fail…

Change can involve experimentation

When **unsure of the outcome**, make it **clear in advance**

Continuously **learn to adapt**

Choose when failure is acceptable

Thank you and questions

Thank you and questions

Reach out to me:

Email: jeremy@jeremycook.ca

Twitter: @JCook21

Thank you and questions

Reach out to me:

Email: jeremy@jeremycook.ca

Twitter: @JCook21

Any questions?