

You're an Architect... Now What?

Nathaniel T. Schutta
[@ntschutta](https://twitter.com/ntschutta)

Many developers
have a similar goal.

They want to make the
tough technical decisions.

They want to be an architect!

The path to becoming an architect is varied.



And many of us climb
up that mountain.

Only to get there and
go...this is it?

We may think we'll have
complete authority....

But it usually isn't
quite that simple!

Have to manage
competing agendas...

Many of the skills that got
us there won't keep us there.

Or won't guarantee
raises and bonuses.

The key is adapting.

Staying agile.

Embrace new challenges.

It's a different world!

Not always what
you expect...

But it can be very rewarding.

It isn't what you think it is.

Many of us love to program.

It's fun to build, to create.

It's a very objective thing.

Code compiles or it doesn't.

Tests pass or they don't.

Work comes in a
(relatively) steady stream.

Pull the next card.

Unless there is a fire drill...

Architecture doesn't have
a red/green bar.

Well, not exactly.



Coding is often
instant gratification.

Or instant frustration...

Architecture doesn't have
that property.

Payoff takes months,
even years.

But it is special
when it happens.

You'll probably still code.

Despite what some
seem to think!

But a lot less than before.

Are you OK with that?

You're expected to lead.

Be a change agent.

Introduce new ideas.

You'll usually have
more autonomy.

Tends to be less task based.

At least they aren't always
coming from a backlog.

May need to go *to* the
work, it won't come to you.

Tasks are often more
nebulous in nature.

Evaluate X.

How?

Figure it out...

You'll likely have to go to
more meetings.

Sorry about that.

And get ready for
budget season.

May be in more of a
governance role.

May need to tell people NO.

But in such a way that
they understand why ;)

Explain the strategy,
roadmap, vision.

It is what you make it.

Can go in many directions.

Self motivated.

Unlikely to have a PM
telling you what to do.

Are you working on
the right stuff?

You better know!

Time management
can be an issue.

You'll likely juggle multiple priorities at once.

Be ready for the one offs...

And the ad hoc “can you
help with this”?

Be aware of “growth opportunities”.

Sometimes this is code...

For something your boss
doesn't want to do.

Can be like broccoli...

May not like the taste, but
it's still good for you.

Visibility matters.

Do your customers know
what you are doing for them?

Are they happy?

Technical skill is assumed.

Raises and bonuses
come for other reasons.

Technical problems?
Hire an engineer.



Who you gonna call?



How about now?

Fixing a pipe? Need
a new outlet?

Tearing down walls?
Adding a new room?

It isn't just about technical
skills anymore.

Business, technical,
communication, leadership.

How well do you know
the business domain?

Do you have technical
breadth and depth?

How well can you
communicate?

To which levels?

Lead yourself, direct a
team, influence an org?

Role dictates what
proficiencies you need.

VP of arch probably isn't
as technical as a SA.

But strong leadership and communication skills.

What are your strengths?

What do you need
to work on?

Take advantage of
educational opportunities.

You'll spend more time
communicating.

In meetings, over email,
creating documents.

Often directly with people.



You won't like all of it.

Some of it, frankly, isn't
very enjoyable.

Or as your boss says: “it comes with the paycheck”.

End of the day, it is a more subjective experience.

You'll do more things that
don't compile.

Are you OK with that?

Because if you aren't...

You may want to rethink
your career goals.

Communication.

Communication skills are
vital for an architect.

Presenting, architecture
documents, email.

“Architecture” is essentially
a fiction, a story.

Can you tell that story to
your developers?

If those implementing
it don't grok it...

They can't build it.

The team should get it.

Can they explain it?

They should be able to tell
the story too.

So have them.

Standups, retros, etc.

But it isn't just a matter of
implementation details...

Must be able to explain it
up and down the org.

What about the people
that sign the checks?

They won't be impressed
by spewing acronyms.

Learn from sales engineers.

How are products marketed?

To whom?

Some vendors speak exec
better than others.

Take notes...

How you communicate
matters more than you think.

Banish RTFM from
your repertoire.

Presentations matter!



Presentation **PATTERNS**

Techniques for Crafting Better Presentations



NEAL FORD | MATTHEW MCCULLOUGH | NATHANIEL SCHUTTA



Presentation Patterns.

<http://presentationpatterns.com/>

What is most effective in
your organization?

What style works best?

Embrace that...even if you
have to plug your nose.

What is the pain point?

Can you tie things back to
exec level goals?

Even if it's a stretch...

Sonar, Jenkins.

Listen.

What are the concerns?

Some legitimate issues.

Don't just dismiss them!

Address them.

Have a conversation!

This won't work...



How are you
communicating?

With respect?

With derision?

What works better?

Be ready for the game of
“enterprise operator”.

“I heard X said Y is our
enterprise standard”.

“Bob said we’re out of
compliance with...”

You can't stop that.

Especially in large
organizations.

It ^{*}will^{*} happen, be prepared.

Can you tell the story?

1, 5, 30 minute.

What would you say to
the VP in the elevator?

What would you say at a
team meeting?

What about at an
SLT meeting?

What if you were asked
for an overview?

Practice it!

You will repeat yourself.

“Say it 3 times”.

Consider an architectural
concept document.

One or two pages.

Key concerns, gaps,
capabilities, etc.

High level.

Whatever helps you
tell the story.

Influence.

How do you get people to
do what you want?

Two approaches...





We can't force change.

But we can influence.

Find the influencers.

Keep in mind:

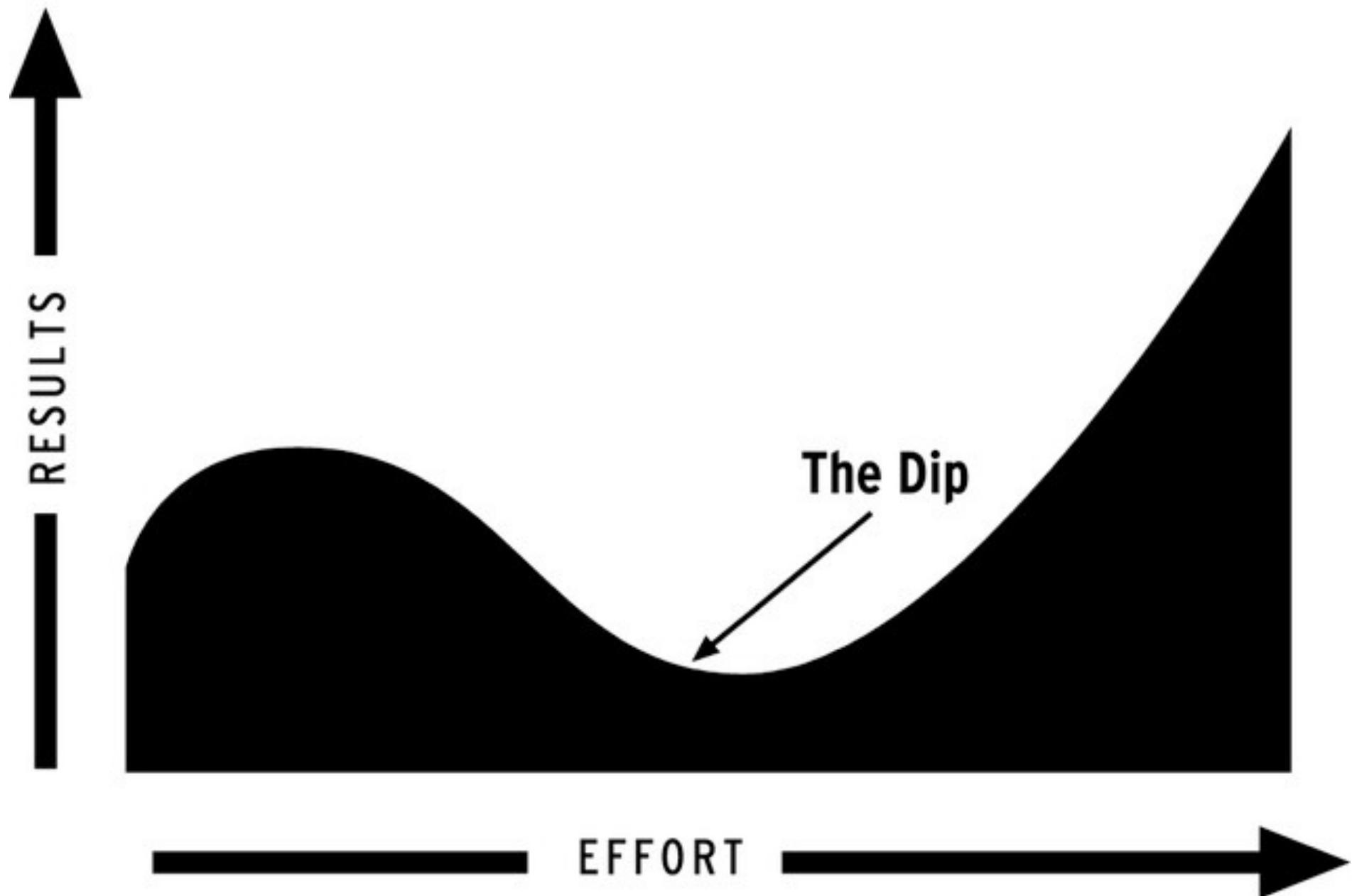
“Change must happen with
people not to them.”

— David Hussman
<http://devjam.com>

Change is hard to maintain.

We get worse before
we get better.

It's easy to back slide.



http://sethgodin.typepad.com/the_dip/2007/05/images_from_the.html

The Dip.

http://sethgodin.typepad.com/the_dip/

We have to push
through the dip.

Thus why force
doesn't work well...

Change takes time.

Be patient.

Change can overwhelm.

Don't try to change
too much at once.

Recipe for failure.



Crawl, Walk, Run.

It's not easy!

There's no recipe.

Stakeholder management.

Be acutely aware of
the stakeholders.

How are you perceived?

What do they say when
you're not there?

You better know!

Be proactive.

Drop by, call, email them.

Don't assume
silence is golden.

Engage with them.

Process isn't as important
as engagement.

You get credit for what
they know about.

Be sure they know what
you're doing for them.

May not be able to directly
access senior people.

Especially if they don't
know who you are.

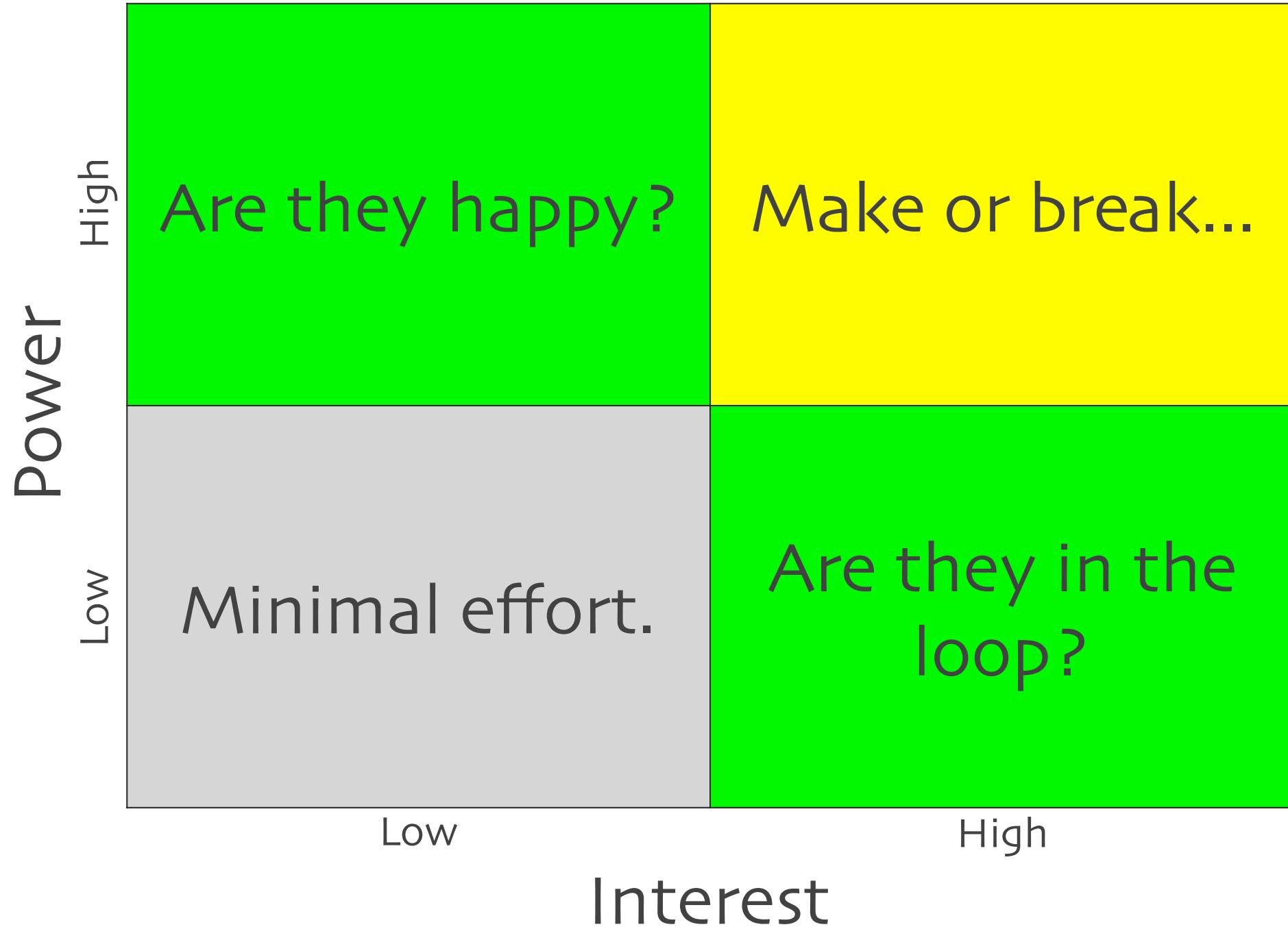
“Colonels don’t talk
to captains”.

They have trusted
sources...who are they?

Work with them.

Leverage meetings especially
if their boss is there.

Not all stakeholders are
created equally...



Low interest? Probably
not a focus area for you.

High power? High
interest? Key people.

Make sure they get
what they need.

Must understand
your stakeholders.

What's important to them?

What's their background?

What are their concerns?

What is the decision
maker influenced by?

Who has her ear?

Influence the influencers.

“A reform often advances
most rapidly by
indirection.”

— Ms. Frances Willard

Approach as equals.

Rely on the strength of your
ideas and your reputation.

Find common ground.

Reciprocity rules...

Be helpful.

Be respectful.

Research your ideas.

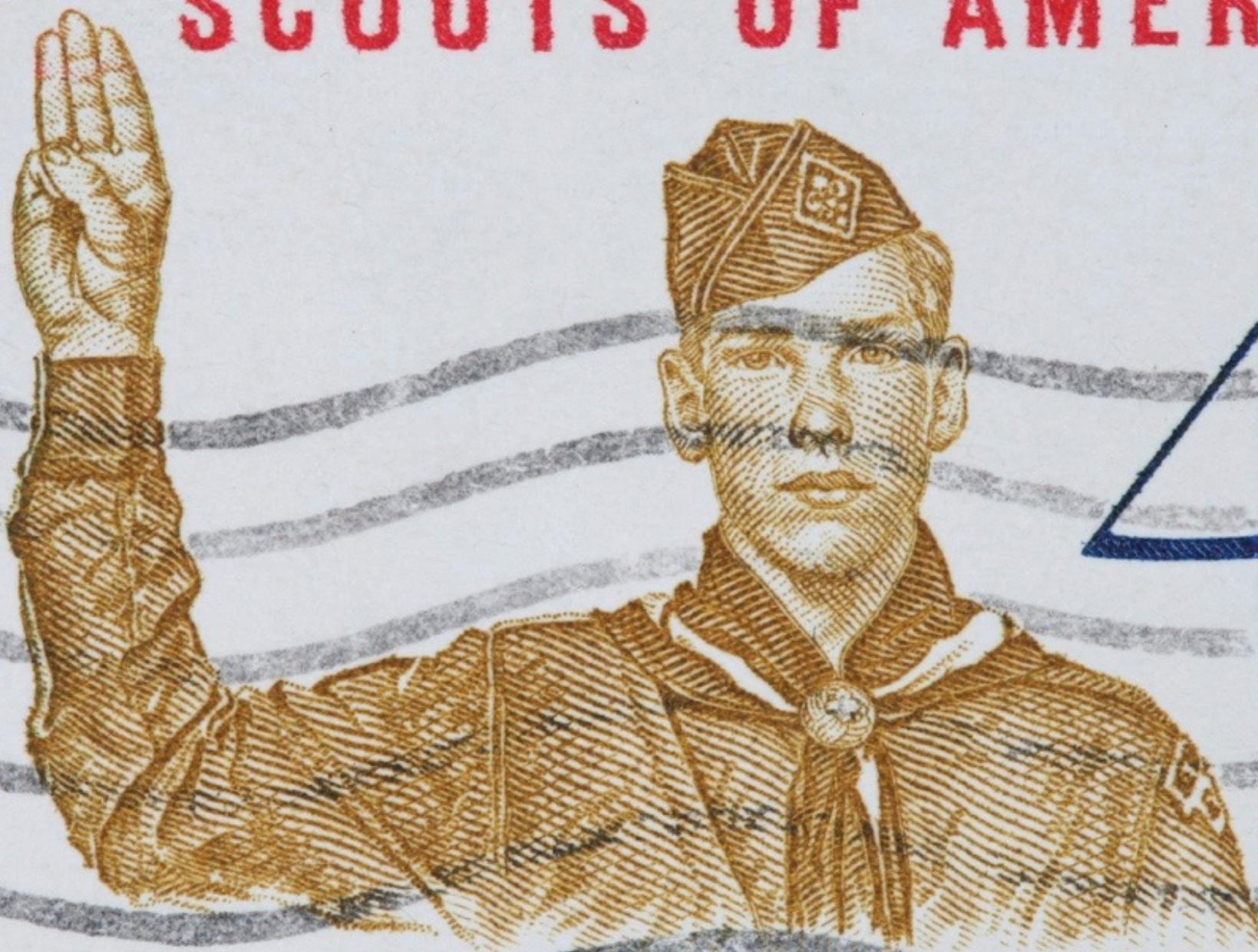
Use trusted sources.

Recruit credible allies.

Consider things from your
manager's point of view.

BOY

SCOUTS OF AMERICA



4c

U.S.
POSTAGE

Be prepared.

Have your story ready.

There might be a rational
reason for current state.

Maybe.

What I told you was true...
from a certain point of view

Over thinking it will
drive you nuts.

Tech is easy.

Culture is hard.

Speaking of cultures...
what is yours like?

What does your
company value?

Innovation? Stability? Cost?

How is technology viewed?

Strategic asset? Cost center?

Culture trumps all.

And it is *really*
hard to change.

“If they don’t change the
paint once in a while...”

Be aware of your culture.

Can you work within
those constraints?

It will inform virtually
everything you do.

Relationships.

It isn't just about the tech.

Relationships matter.

It's about your network.

Who can you reach out to?

Who knows you?

Must move beyond your
immediate peers.

Retail politics.

Requires cultivation.





Make your coworkers
feel valued.

Show interest in them !

Be empathetic.

Golden rule...

Be respectful.

Say thank you.

Be genuine.

Compliment people.

Be positive.

Ask for and offer help.

Active listening!

Understand their
work style.

Volunteer.

It isn't that hard...

Grass roots.

Make your work visible.

Sometimes the one off
tasks reach far and wide.

what's next?

Architect is like
developer or designer.

It's a broad term.

Means different things to
different people...

And across organizations.

Like “developer” there can
be many roles.

Solution Architect.

Direct project responsibility.

Day in day out with a
project team.

More coding.

Live in the project room.

Often a senior developer.

Most people start here.

Portfolio Architect.

Handles a set of
applications.

Works with the SAs.

Involved with initial stages
of a project and reviews.

Helps with thorny issues.

Tend to work more with
the business leadership.

Vision and roadmap.

Usually high visibility.

May have direct reports.

Domain Architect.

Responsible for a
technology or capability.

May work more on
foundational tasks.

Mobile, Java, Security,
Integration...

Reference implementations,
standards, best practices.

Evaluate new technologies.

More proof of concept
type programming.

Work with the PA to
identify gaps, priorities, etc.

Enablement, help
teams succeed.

Not (just) about making a
decision on a library.

Making sure teams have
what they need.

Discipline Architect.

Focus on specific aspect of
technology or business.

Information, security,
business domain.

Consult on a variety of
projects across the org.

Enterprise Architect.

Works across an organization.

Often very senior.

Identify patterns,
roadmaps, vision across org.

Should aggregate up from
project/portfolio/domain.

Drive enterprise
level solutions.

Often work with
senior leadership.

May have little to no day
to day project interaction.

VP of Architecture.

Far less technical in nature.

More organizationally
focussed.

Managing budgets, egos,
developing staff.

These aren't static.

You may span roles.

Depends on the
organization.

And you can move from
arch to other roles.

Will you grow technically?

Yes - but not as much.

Focus will be on leading
and influencing.

Bigger projects, larger teams.

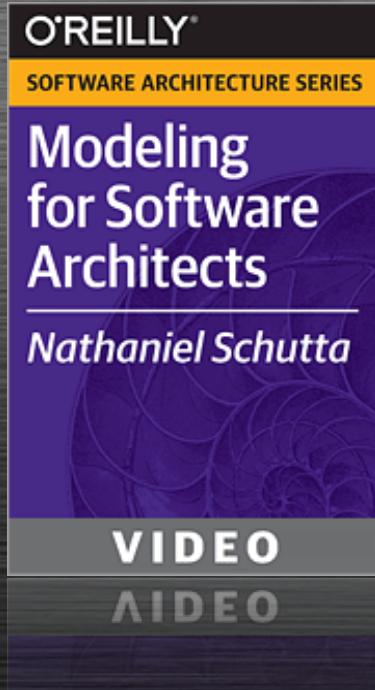
Being an architect...may
not be what you imagined.

But it can be a great job!



Have fun!

Thanks!



I'm a Software
Architect, Now
What?

Nathaniel T. Schutta
@ntschutta