Leadership & Influence

How to bring the x-factor into your team

James Hou @ Norcal Dreamin’ 2019-06-27

Overview

1. About me and my story
2. Leadership principles & traits
3. Communication strategies & keywords
4. How I drive change for common challenges
5. Re-visit, deep dive or open-workshop

https://bit.ly/2XdNy6b

About Me

Psychology major switched to B.A. in International Business.

Self-taught - hobbyist techie to **problem solving** professional.

8 years in ecosystem. Accidental Architect @ Colliers => Independent Consultant @ Google.
Startups, big firm consultant, freelance, digital nomad, **tiny** & **large** enterprise teams.

Embraces both **business** and **technical** lens of the ecosystem/platform.

20% Business Analyst / Scrum Master (Project Mgr lite) / Product Owner.
80% Architect / Developer.

My Story (1/3)

@Colliers

Took sole ownership of 150 user SFDC Org **2 weeks into job** after my mentor left.

Self-advocated for space to learn and drove a rough Software Development Lifecycle (SDLC) to meet demands from small team.

My Story (2/3)

@Genentech
Always given the choice of which projects due to “leadership” skills. Took on the impossible projects and drove them to completion on schedule.

Business leaders and implementation teams respected my communication and decisions. I was the **go-to**.

Championed **innovation** projects and a **proper SDLC** within the implementation team. Eventually, lead and drove org migration / refresh ($$mm) project which required change management coaching, influencing, storytelling, and prototypes **all to gather buy-in**.

My Story (3/3)

@Google

Independent Consultant (Non-FTE) given **product ownership** and **champions/drives** an **SDLC** of my own design.

I am one of the driving forces for all technical Salesforce roadmapping for Google Play Gift Cards.

I listen, influence, and **drive what I believe is right** for both the technology and business teams.

Leadership Principles

“It’s ok to fail.
Pick yourself up, clean up the mess, try again.”

Don’t be afraid to fail. Keep moving forward.

Foster resilience

Not being afraid of failure **reduces the cycle (and pain)** of repeating the learning loop. Attempt, fail, learn, repeat. The faster this loop, the more resilient you become.

No risk, no reward

Success is built atop failure. Always operating in a comfort zone means you’re not taking risks. **Not taking risks leads to stagnation** of skills, growth, and innovation.

Earn trust, earn transparency

Mistakes are ok. Create opportunities for people to fix their mistakes and they will be willing to speak up when something goes wrong, not hide it.

> Don’t leave a trail of bodies behind you.

David F.
Head of Patient & Physician Solutions @ Genentech

Bring everyone with you.

Find value in others

Everyone has **something** to **contribute**. You may be an expert at X, but not Y and Z. Don't leave them behind, empower them to help them succeed.

You are what your team is

Lift up everyone around you. Acknowledge and give recognition as deserved, even for the small things. Help those around you and it will pay itself back in the long run.

Be kind

There are no bad ideas. **Everyone has a voice**, and even if it sounds like a bad idea there may be seeds of inspiration. Walk through ideas (good and bad) through **litmus tests** or "**5 whys**". Be patient.

“Is there anything I can do to make your life easier?”

David F.

Head of Patient & Physician Solutions @ Genentech

Be the buffer. Be the unblocker. Be the go-to.

Help your colleagues

If it costs you 10 minutes, but costs someone else 10x longer this is an easy win to **build rapport** and **credibility** as an expert.

Help your juniors

Buffer the noise. As they're learning the ropes, allowing them to get up to speed quicker means your overall team can benefit sooner.

Help your team

In positions of leadership, becoming an umbrella and **protecting the team** is important. **Remove blockers** for them to make their lives easier.

The other side of the coin is to **help them grow** in the direction they are passionate about. **Create opportunities** for everyone to move toward their goals.

Leadership Traits

Connects the dots

Being able to **connect loose ends** and who to talk to to get the **full picture** is invaluable.

This lets you be a trusted resource that **glues** various **business units** or teams together.

This also lets your solutions paint a more **complete picture**, since only you can account for edge cases that come up when 1+1+1+1 occurs over a project cycle.

Manages time well

The ability to understand execution of small tasks that **add up of the overall landscape** of your week, month or quarter is paramount.

Deflect, refuse, or negotiate better outcomes (time and/or tasks) to position your team for success.

Take charge of pushing forward agendas that make significant impact. Trust in the **Pareto principle** (80% outcomes come from 20% of the effort).

Budgets for 20% time

It’s important to take your mind off the **day to day**. Build in some side projects, time for learning something new, or even conferences out of industry.

Coming back with a **fresh set of eyes** on something you’re working on can boost productivity - sometimes you need to just **think about something** else.

Generalist and specialist

Be attuned to **cross-discipline skills** and **ideas**. You never know where inspiration and creativity will stem from.

Things you learn from hobbies, or from a different discipline (project management, software quality testing) can **influence and reshape** how you think about your day-to-day.

Great communicator

Every “natural” leader I’ve come across has actually **honed a set of communication skills**. The way they command the room is, in reality, an entire suite of sub-skills.

Being **transparent** and **precise** in their communication makes them “easy” to work with.

Communication

Anchoring

Initial point of reference

- Usually an agenda, goal, or action items desired.

Sets tone

- Similar to writing stories / narratives, this sets tones of the coming conversation.
- Always anchor what a meeting is (what outcomes you want out of the meeting).

How

- Stating action items desired at beginning of meeting / email - then set context.
- When conversations de-rail, anchor back to the original focus.
- In high level conversations, **avoid deep dives** because the focus was to drive ideation / feasibility, not solutioning.
- In deep dive conversations, **avoid tangents** and focus on the original solutioning design.

Examples

- “I **like** where this conversation is going, but **can we table that** to explore later? I want to make sure we [insert current goals].”
- “I am blocked by X, I **need** Y **to** move forward **so that** we can Z. Here is the current situation.”
- “This meeting **is about** [ideating around X, getting clarity around action item Y].”

Context and Progression

Ongoing reference

- Gives background to why time is being spent on this conversation.
- Can provide the “why” or the “so what”.

Make connections

- Like a 5 paragraph (MLA) essay, transition your points cleanly.
- The better the context and progression, the better they support your ideas.

How

- Progress context **forwards**. It’s easier to follow a linear timeline and visuals help (prototypes, wireframes, demo-ing UI, whiteboarding etc).
- When discussing **nouns** (proper nouns or industry / technical nouns), **repeating context for new individuals** or if there has been a gap in time for participants.

Examples

- “During the course of A, B and C, we are now stuck at C. I **believe** the best option is D so this is what I **propose**”
- “The SAA, the Super Awesome Application, is…”
- “I’ve invited Duke, he helped [on this project or during that process] **a while back**, to help us on…”

Expectation Management (1/3)

Be Realistic

- Infeasible solutions or aggressive timelines which lead to burnout doesn't help anyone. If something doesn't feel right or you don't know if it's possible...

Be Truthful

- **It's ok to push back, say no, or say you don't know.** This builds credibility. Saying you don't know, but asking for time to research is more valuable than answering yes and running into a dead end.
- When pushing back or giving a negative response, **provide alternative paths** forward.

Set Priorities

- **If everything is P1 nothing is P1**.
- Keep a **Business Stack Rank** and a **Sprint Stack Rank**.
  - Business owns the overall ranking. The fewer the decision makers, the better.
  - Implementation team takes it into the work queue as bandwidth allows.
- Some bandwidth is **always reserved** at discretion of implementation team (innovation, tech debt removal, extra testing, more business features etc).

Practice LOE

- The hardest skill is **solutioning-on-the-fly**. Knowing something is possible and the rough level of effort (LOE) and bringing that skill to strategy or high level solution meetings is a game changing skill. However, it typically requires previous (or current) hands-on skills.

Expectation Management (2/3)

Priority LVL

- P1 Bug - **No workarounds** OR workaround is **excruciatingly painful** and unacceptable. **Hotfix** now.
  - Feature Request: Automation would provide **significant** impact.
- P2 Bug - Viable, **painful workaround** exists. Fixed **next release** window.
  - Feature Request: Automation would provide **great** impact.
- P3 Bug / Feature Request - Nice to fix/have.
- P4 Bug / Feature Request - Cosmetic.

LOE Rubric (Points)

- 1 - 30 mins.
- 2 - 30 min to 1 hour.
- 3 - 1 hour to 4 hours.
- 5 - 1 day to 3 days.
- 8 - Minor discovery, 3+ days.
- 13 - Large discovery. Break into 8 and 5. Or, keep as 13 for an **EPIC**.

Expectation Management (3/3)

How

- Establish a **backlog**, somewhere to collect centralized feedback.
  - Gain **champions, decision-makers** who can help groom this backlog.
  - Remember - business owns one ranking, implementation owns a second.
  - Establish a **cadence** which to tackle **discovery, assessment, execution** and **communication** of backlog items.
- Route requests to the communication channel(s), but execute **quick-wins** if they are **config** only.
- Solutioning-on-the-fly and take a **first-pass** to establish baseline feasibility (0-50% confidence).
  - When **prioritized**, complete **discovery/sizing** from bottom-up or top-down.

Examples

- “We need a **method to the madness** and I need to start keeping track of all this. Can we establish a funnel for this? I’ll **drive this** but I’ll need X, Y and Z’s **support**.”
- “Do you mind **documenting a request** for this? I’ll be able to keep track of things better there.”
  - “If you can’t**, no worries**, I can log it for you add you as the **requestor**.”
- “My current **bandwidth** is **maxed** out, if you don’t want to risk my **current priorities**, can we document it and I’ll take a look at a better time?”

Empathy

Agendas

- Promotion? Above/below the radar? Designing a process? Everyone has things they're working on.
  - Keep communication **at the level** they care about.
  - Know who they **report to**, and what their agendas are.
- This will set **context for you** on if they need to be looped in. It also informs you who to **influence**.

Mentoring

- Everyone was a newbie **at one point**. Place yourself in their shoes. Be patient.

XY Problem

- Technique to **re-anchor** a conversation to deduce what the **actual issue** is.
- Used frequently for **problem solving** during customer support calls.

How

- Ask people how you can unblock them, how can you help them.
- Ask them what they're working on.

Examples

- "I **know you're concerned** about X and Y, so I've tried to address this by Z"
- "Let's start **from the beginning**, can you walk me through the original problem?"
- "It seems like we're trying to address the **downstream symptoms**, is there something earlier that could be causing this issue? Let's solve **for that upstream root cause**."

Know Your Audience

Catering

- Know the goals of the audience you are communicating to. Speak **at their level**.
  - In leadership meetings, address **high level** costs:
    - Cost of build, cost of ownership, timelines, and flexibility.
  - In solution meetings, address **low level** costs:
    - Tech-debt, correctness, edge cases, and user experiences.

How

- Hold **focused** meetings where **action items are requested** of decision makers in the room.
  - FYIs can be emails or in collaborative documentation.
- **Anchor** topics and **context** only on pertinent points as you **make your argument** toward **asks**.
- In sync/report-out type meetings, **succinctly** report the level of findings your audience cares about.
  - Modulate **high detail vs low detail** depending on audience composition.

Analogies

KISS

- Part empathy, part good communication - **keeping it simple** allows the entire audience to follow along.
- **Avoid technical jargon** unless you know it will be understood.

Relatable Anchoring

- Technology is abstract. Abstract things are hard(er) to **wrap your head** around. Concrete analogies are **easier to relate** to.
    - Cars: makes, engines, racing, maintenance (oil, tires), big repairs (transmission, engine).
    - Houses: foundations, wiring, extra rooms, maintenance, remodel, rebuild.
    - Hobbies etc.

Examples

- “Addressing **technical-debt** is like **changing the oil** and **tires** on your car. Don’t do it for a long time and you can say goodbye to the **engine**.”
- “The **foundation** we build X projects ago wasn’t designed to **support the weight** of feature Y. However, we could possible do this Z instead…”
- “In Formula 1, **brakes** aren’t used to slow the car down, they’re used to **set up** the **maximum speed** to go around the corner **without crashing** into the wall.”

Negotiation

Options

- Never come empty handed. Provide **alternatives** for what is being negotiated out.
- Sometimes choosing a **painful** option as a solution is a valid strategy to **influence user behavior**.

Risk

- All options have levels of risk, it's important to detail out the **what** they are and **why** the risk exists.
- Identifying risk levels (technical, business process, unknowns, teams, cross-teams) provide more data points to **successfully identify alternatives**.

Evidence

- Part of establishing **credibility** is providing supporting evidence.
  - Credibility is established through **accuracy of sizing** and **delivering on time**.
  - Without established credibility, in a technical negotiation, usually only senior or external expertise can provide enough **weight** to tip scales.

Examples

- "I don't think that's feasible, but **what we can** do is…"
- "Option A has **less risk and unknowns**, however, option B is **overall better solution** with **more unknowns**. We can go with option A if I can **refactor it next sprint**, but we can go with B and **possibly defer feature ZZ** if I need more time."

Influencing & Driving Change

Driving Change

Establish Credibility

Managing expectations, timelines and feasibility assessments from your asks and then executing them on time.

Calling risks out as you see them and managing the timeline reassures what the **realistic** deliverable trajectory is

Find champions at your peer and superior level to help you on this journey. Make your case why what you're trying to establish is important.

Promote Transparency

When gaining champions for your cause, be **absolutely truthful** - even if it means calling your own shortcomings or identifying **risks of taking on extra work**.

Good managers and leaders will be understanding and **protect** you.

Open communication (positive and negative) is what **earns trust** and builds increased credibility.

Under-promise, over-deliver

The human brain isn't meant to operate on **crunch** for prolonged periods of time.

Build in **buffer** into all your assessments for the unknowns. There are both **technology** and **people** unknowns.

As familiarity for both **increases**, **decrease** (but never remove) the buffer. Heroes are not sustainable. Greek heroes usually perish.

Common Pains

1. Backlog, method to the madness, user stories
2. Bandwidth (or lack thereof)
3. Technical debt avoidance and removal
4. It’s the same thing every week, every quarter...

Driving Forward an SDLC

1. Manage **your own work queue** in a transparent manner.
2. Begin putting the cycle of **discovery, assessment, delivery** and **feedback** into a documented, transparent and communicable **backlog**.
3. **Influence champions** on the "*so what*" and **invite** conversation and feedback loops during this process.
4. Showcase **improvements** in delivery, accuracy, and accountability in the overall implementation process.
5. **Identify more champions** (e.g. product owners) who can chime in on **improving the system**. More end-users, management, leaders can **track system improvements** from **idea to delivery**.
6. **Defend the backlog** priorities, your or your team's bandwidth, and the overall process.

Managing Bandwidth

1. **Establish credibility** to deliver, with or without an SDLC. However, establish an **LOE sizing** and **feasibility** metric.

2. Be accurate with those metrics, establish **transparency** and **trust** in that you're accurate and believable.

3. **Identify** champion(s) who are in charge of your priorities. **Influence** them that you can only take on so much. **Communicate** comfort zones and risks for stepping outside those.

    - "I work best with only **two major context switches per day**. More than that and productivity and **quality will suffer**."

4. Always create a **buffer** when dealing with:
    - Unknowns in tech (<90% confidence on feasibility).
    - Unknowns in people (new resources, other resources bandwidth constrained).
    - Unknowns in the business process (discovery required).
    - When working with Salesforce tech. Respect the "gotchas".

5. Reduce the **buffer** as familiarity with each of the above increases. Always **maintain a buffer**. See LOE rubric.

Tech Debt - Avoidance

1. In **strategic meetings** where blue-sky ideation is happening, be **open-minded to all options** on the table.

2. Assign **sizing**, regardless of the ask. Requests that up-end or break **foundations** are not inherently bad, they just require **proper sizing** to include all the risks to avoid accrual of debt during **delivery**.

3. It is perfectly acceptable to **knowingly take on debt** if there is a **process for removal**. It's like not changing the engine oil knowing you've made an appointment in 2 weeks.

4. Depending on the **composition of the technical team**, debt means **clicks OR code**.
    - Best practice **is different** per business group and technical team and **different per project**.

5. **Consistency** in implementation (naming, style, solutioning) is a strong way **avoid some kinds of debt**.

6. **Agile**-ish (Agile-fall) SDLC is a good way to quickly iterate **MVPs** to test your **solution hypothesis** before building too deep.
    - It's also a good way to have a quick **feedback loop to remove technical debt**.

Tech Debt - Removal

1. Establish and influence an **20% bandwidth hold** rule.
   - 80% of a deliverable cycle is reserved for **solutioning**.
   - 20% is used at the implementation team's discretion (**debt removal**, **innovation**, or **more features**).
2. (Re) establish **consistency**.
   - No declarative on mission critical objects, only one PB per object, one PB per event per object, when doing X, always do Y, etc.
3. **Negotiate** the importance of maintenance using **analogies**.
4. Ensure **unit** and **integration** (behavior) tests are working to spec prior to a refactor.

Injecting Innovation

1. Utilize part of the **20% bandwidth hold** to do interesting things every now and then.
2. Create **prototypes** or **show-and-tells** which are experimental in nature to **drive conversation** in either the technical or business teams.
3. **Gather feedback** from those conversations and either **iterate or pivot** those prototypes.
4. It's also okay to use this time to **practice or utilize new technologies** since the landscape of both tech and Salesforce is ever changing.
5. **Bring feedback, techniques, or novel solutions** back into other parts of your stack or derive new features that come out of this exploratory effort.
6. **Socialize** the outcomes of these efforts. **Creating a space** to do this **opens opportunities** to invite communication and ideation. Lead that space.

Wrap-up

1. Communicate clearly and transparently
2. Manage expectations of yourself, of the team
3. Find champions for what you care about
4. Drive the cause you want to change
5. Reserve 20% of team bandwidth

tsalb.github.io/ncd/leadership-influence-2019/slides.pdf

Thank You