



# Nail the Pitch and Stick the Landing

How To Propose & Deliver Major Technical Projects

---

# Ei-Nyung Choi

she/her

Fractional CTO / Startup Advisor /  
Engineering Leader

ex-Slack, ex-The Browser Company



---

# Ei-Nyung Choi

she/her

Fractional CTO / Startup Advisor /  
Engineering Leader

ex-Slack, ex-The Browser Company



---

# Ei-Nyung Choi

she/her

Fractional CTO / Startup Advisor /  
Engineering Leader

ex-Slack, ex-The Browser Company





# A recent story



**Excitement**



**Excitement**



# Apprehension



---

**A theoretically easy task**

---

**Reality: a *lot* of manual work**

---

**Reality: a *lot* of manual work**



# The real schedule

---

# My first (unsuccessful) pitch

---

# My first (unsuccessful) pitch



# Building frustration

---

**Egregious example**



---

**A familiar story**

---

# A familiar story

---

# A familiar story

**Why can't everyone else see how important it is?**

---

**Make it easy and safe for the  
decision-maker to say yes**



---

# Common mistakes



# Common mistakes

- You are centering the wrong person
- You are trying to convince others to adopt your priorities
- You are not clearly showing the cost/benefit
- You are not addressing their worst fears
- You don't have a clear execution plan with escape hatches



How?





# How?

By answering these questions:

- Who can say yes or no to this project?
- What is that person's highest priority right now? What goals are they held accountable for?
- What is their biggest nightmare? What keeps them up at night?

---

**Who can say yes or no?**

---

**What is their highest priority?**

**Don't try to change their mind**





# Find the fit

- Does your project fit their priorities?
  - ◆ If yes, is it easy to see the connection?
  - ◆ If no, it may not be the right time

---

**What keeps them up at night?**

# Nightmare A



# Nightmare B







# Review

- Who can say yes or no to this project?
- What is that person's highest priority right now? What goals are they held accountable for?
- What is their biggest nightmare? What keeps them up at night?

**Make it easy and safe for the  
decision-maker to say yes**



---

# Structure of a pitch



# Structure of a Pitch

- Problem statement
- Cost-Benefit Analysis
- Risks & Risk Mitigation
- Open Issues



# Problem statement

- Where does the problem exist?
- What is the business impact of the problem?
- How does the problem impact team/department/company prioritization?



# Costs to *them* in terms of *what they value*

- How much will it cost to leave the problem as-is or let it grow?
- How much will it cost to let you take on this project?



# Benefit to *them* in terms of *what they value*

- How much will it benefit them to leave the problem as-is or let it grow?
- How much will it benefit them to let you take on this project?



# Example cost-benefit breakout

- **Current cost** of turning on a new locale (no customizations): 0.5 days of developer time + 5 days for sign off & release, 4 repos touched
- **Proposed cost** of refactor: 1 developer day + 7 days to get sign-off from other teams & release, 4 repos touched
- **New cost** of turning on a new locale: 1 hour of developer time + nominal release overhead, 1 repo touched
- **Net reduction** in time: >90% reduction in turnaround time





# Example cost-benefit breakout

- **Current cost** of adding keyboard support for a new locale: 2-3 days of developer time + 1 day for sign off & release, 18 files touched
- **Proposed cost** of refactor: 2 days of developer time + 1 day to get sign-off from other teams & release, 22 files touched
- **New cost** of adding keyboard support: 1-2 days of developer time + nominal release overhead, 2 files touched
- **Net reduction** in time: >50% reduction in turnaround time



# Risks and risk mitigation

- What can go wrong? Is there a rollback strategy?
- What if we invest in the project but it doesn't improve things?
- What if this project goes on forever?
- What if this breaks the product?



# Open issues

- What unanswered questions do you have?
- What moving dependencies does this project have?
- Have a plan
  - ◆ Who will you talk to?
  - ◆ When will issue be considered closed?



# Execution Plan



# Execution Plan

- Communicate, communicate, communicate!
  - ◆ Kick off, status reports, mid-project check-in, dashboard, etc. Never wait to be asked
- Prioritize the biggest bang for buck first
- Release incrementally with immediate benefit in each release
- Be ready to stop the project at any point
- Aggressively cut scope at every stage

---

# Contact info

- <https://www.linkedin.com/in/einyungchoi>
- <https://calendly.com/eingy/intro>
- Get the deck:

