

CSE 403

Pitching a Project Idea

Reading:

Pragmatic Programming Ch. 1, 10

slides created by Valentin Rasmov and Marty Stepp

<http://www.cs.washington.edu/403/>

Lifecycle Objectives

- **Operational Concepts**
 - What is it? (**High-level scope and objectives**)
- **System Requirements**
 - What does it do for us? (**Lower-level actual planned deliverables**)
- **System and Software Architecture**
 - How? (**Technically**)
- **Lifecycle Plan**
 - Who wants it? Who'll support it? (**Resources needed**)
- **Feasibility Rationale**
 - Given the constraints, is this realistic (**can it be built**)?

1. Operational concepts

- **Top-level system objectives and scope**
 - What problem are you trying to solve? Why? For whom?
 - User community, environment, major benefits?
 - Goals and non-goals
 - To set realistic expectations in the audience

Tip: This is what you should be able to explain in a 1-minute pitch (if you didn't have more time) – a.k.a. “an elevator pitch.”

Tip: It takes practice to refine this, so start early.

2. System requirements

- **Essential features of the system**

- What does the customer want from this system?
 - Look from the user's perspective
 - **Tip:** Avoid details at the start; there's time to evolve.
- Discuss main capabilities, outcomes, reliability and performance needs, appearance
- *Customer involvement* is important and beneficial
 - They know best what their interests and needs are, including what fits in their daily work and life patterns
 - ... even if they may not always express it very well
 - They understand the domain better than developers do.
 - Working jointly and openly with customers helps build trust, so any necessary changes are more acceptable.

2: System Requirements

- **Essential features of the system**

- This will be your *initial* written specification
 - Customers can review and sign off or complain early.
 - Putting it in writing is less ambiguous than saying it.
 - Forces you to think of major functional areas and seek architectural defects early
 - “Failing to write a spec is the *single biggest unnecessary risk* you take in a software project” -- Joel Spolsky
- Be concise yet complete
 - People get attached to work even if no longer of value.
- **Tip:** A picture / diagram is (often) worth 1000 words.
- **Tip:** Scenarios and stories help, but avoid being verbose.

3: System/SW Architecture

- **High-level technical description but with enough detail to allow feasibility analysis**
 - Unlike the previous two elements, this is technical.
 - Architectural flaws will only deepen as you go forward, so look for alternatives while it's still early.
 - **Tip:** Try to come up with several (at least 3) alternative architectural designs.
 - **Tip:** Identify clients, servers, major software components, external 3rd party software, and the interactions between them.
 - **Tip:** Pictures say 1000 words.

4: Life Cycle Plan

- **Identify stakeholders and their roles**

- Users, architects, developers, testers, managers, etc.

- **WWWWWHH:**

Why / What / When / Who / Where / How / How

- Objectives: Why is the system being developed?
- Schedules: What will be done, When?
- Responsibilities: Who will do it? Where are they?
- Approach: How will the job be done?
- Resources: How much of each resource?

- **Tip:** Make your best (educated) guess. Some of this will necessarily change. This is *not* a contract.

5: Feasibility Rationale

- **Conceptual integrity and compatibility**
 - Can this really be built with the available resources?
 - Identify project risks
 - What are the assumptions? Any unwarranted ones?
 - “If you make one or two ridiculous assumptions, you’ll find everything I say or do totally justified.”
-- Ashleigh Brilliant, 1671
- **Tip:** Keep asking “why” until the assumptions emerge.
 - E.g., “Why do we need this?”, “Why is this good?”

When do we make pitches?

- To colleagues: to argue for a technical direction
- To management:
 - to convince that your idea/project/approach is of value
- To customers:
 - to purchase/fund product, to change requirements, etc.



Will project be approved?

- Money is rarely the (real) issue...
- The trick is to convince yourself and others
 - that you will deliver a large positive payoff
 - that you can manage the risks while building it
- The different stakeholders (customers, management, developers, etc.) define success differently and you must satisfy (nearly) all of them at once.
- Payoffs: money, market share, credibility, capability, satisfaction
- Risks: wasted money/time, loss of credibility, opportunity cost

Questions audience will ask

- **Q1:** Will this project make a positive difference for us (our company)?
- **Q2:** Do I want to hire this team (to work for us)?

The answers to both are independent, but you'd like both to be "Yes."

Project won't be funded

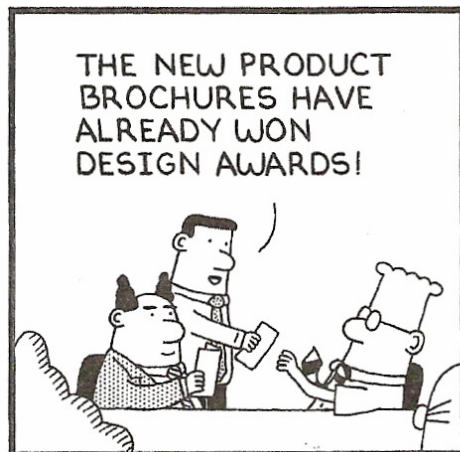
- "... because I think this will be fun to work on."
- "... because it's clearly better technically than the brain-dead solution proposed by those marketing folks who talked to our uninformed managers."

What makes a successful pitch

- Presented at a level appropriate for your audience
 - Customers, marketing people, VPs of Engineering, and developers all have rather different mindsets and measures of success
 - Do you know who your audience will be next week?
- Answering the audience's (unspoken) questions
 - What questions do you anticipate they/we will have?
- Focused and succinct statement of the value you propose and how specifically you intend to provide it
- Credibility of the team
 - If you've successfully completed other similar projects, tell them that, but don't brag.

What makes a successful pitch

- It helps to have something tangible to show briefly (a sketch)
- Distinguish yourself from other same (or similar) products
 - Convince your audience that they shouldn't go to your competition that has already built this product!
- Leave audience with something positive they can remember
- Advice: Keep all of this in mind when applying for jobs and talking to potential employers!



5/30/03 © 2003 United Feature Syndicate, Inc.



Investment process

- create and refine a pitch
- get lots of advice and feedback; refine
- be prepared for detailed diligence
 - market models
 - technical papers and analysis
 - financials
 - customer feedback
- partner meeting
- term sheet
- legal work to close deal

"Elevator pitches"

- 30 seconds to 1 minute in length
 - "Tell me what you've been working on."
- Covers **the most** essential parts from the point of view of your listener
 - No one cares about details and no one will remember them anyway.
- Use your time wisely
 - You are *not* marketing something that's already been made.
 - You *are* trying to figure out if there is a need for what is being proposed.
- A memorable concept to associate with your idea helps tremendously.



Common mistakes

- Not ensuring that everyone can comfortably hear/see you
- Misjudging audience (interests, background, requirements, etc.)
 - E.g., assuming that their understanding prior to the presentation is similar to yours in level of depth
- Not addressing the “why” question to motivate your idea
- Not helping the audience understand the “big picture” of the area in which your product fits
- Not covering alternatives and what novelty you are offering
- Not presenting a picture of how project cost justifies value

Value proposition statement

- Your audience, after listening to your pitch, must be able to *at least* fill out the following template reasonably accurately.
- From “Crossing the Chasm” by Geoffrey Moore

For (target customer)

who (statement of need or opportunity)

the (product or company name)

is a (product or company category)

that (statement of key benefit / compelling reason to buy).

Unlike (primary “competitive” alternative),

our product (statement of primary differentiation).

Value proposition example

For users of the “pine” email client software on Unix
who need to easily find content in their past email correspondence
the “pine+” product
is an email client software
that is backwards compatible with “pine” and also free.
Unlike “pine” or other similar Unix-based email clients,
our product provides an intuitive way to annotate email messages with keywords of the user’s choice in order to facilitate subsequent searching by using one or more keywords in addition to the search functionality that “pine” offers.

Value proposition example

For (target customer)

who (statement of need or opportunity)

the (product or company name)

is a (product or company category)

that (statement of key benefit / compelling reason to buy).

Unlike (primary “competitive” alternative),

our product (statement of primary differentiation).

- This format works for a personal value proposition, as well as for a product value
- An excellent way to distinguish yourself
 - “Unlike the other candidates, I ...”

From "20 Questions for Startup Success"

1. What problem does this product/proposal solve?
2. How does your product/proposal solve the problem?
3. Who are the end users of the product? Is there an addressable market of actual users?
4. Who is your revenue-supplying customer? Are there enough of these paying customers to make a huge business?
5. Does the product solve a problem that end-users/revenue supplying customers actually have?
6. Are you sure that what you're building a product and not just a "feature"?
7. Does your stuff easily fit into the way that people work?
8. Are you too involved in how you're building your product rather than WHAT you're building?
9. Who are the potential partners? Who are the required partners?
10. What is the go-to-market strategy?

From "20 Questions for Startup Success"

11. What is your sustainable competitive advantage? In other words, will you soon be overtaken by others who do something similar?
12. Do you have a time-to-market / first-mover advantage?
13. Can you be #1 or #2 in the space? Who is your competition?
14. Is there a team formed/identified with a record of successful ventures? Have they done something like this before?
15. Is there a "soul" of the team that knows where this product AND business is going for the next few years?
16. Is anyone on the team insane? Also, are the members of the team totally passionate and aligned on this business?
17. Are there product/technology/operations barriers to success? If yes, can they be overcome?
18. Are there marketing/sales barriers to success? If yes, can they be overcome?
19. Are there legal barriers to success? If yes, can they be overcome?
20. Is there an exit strategy?