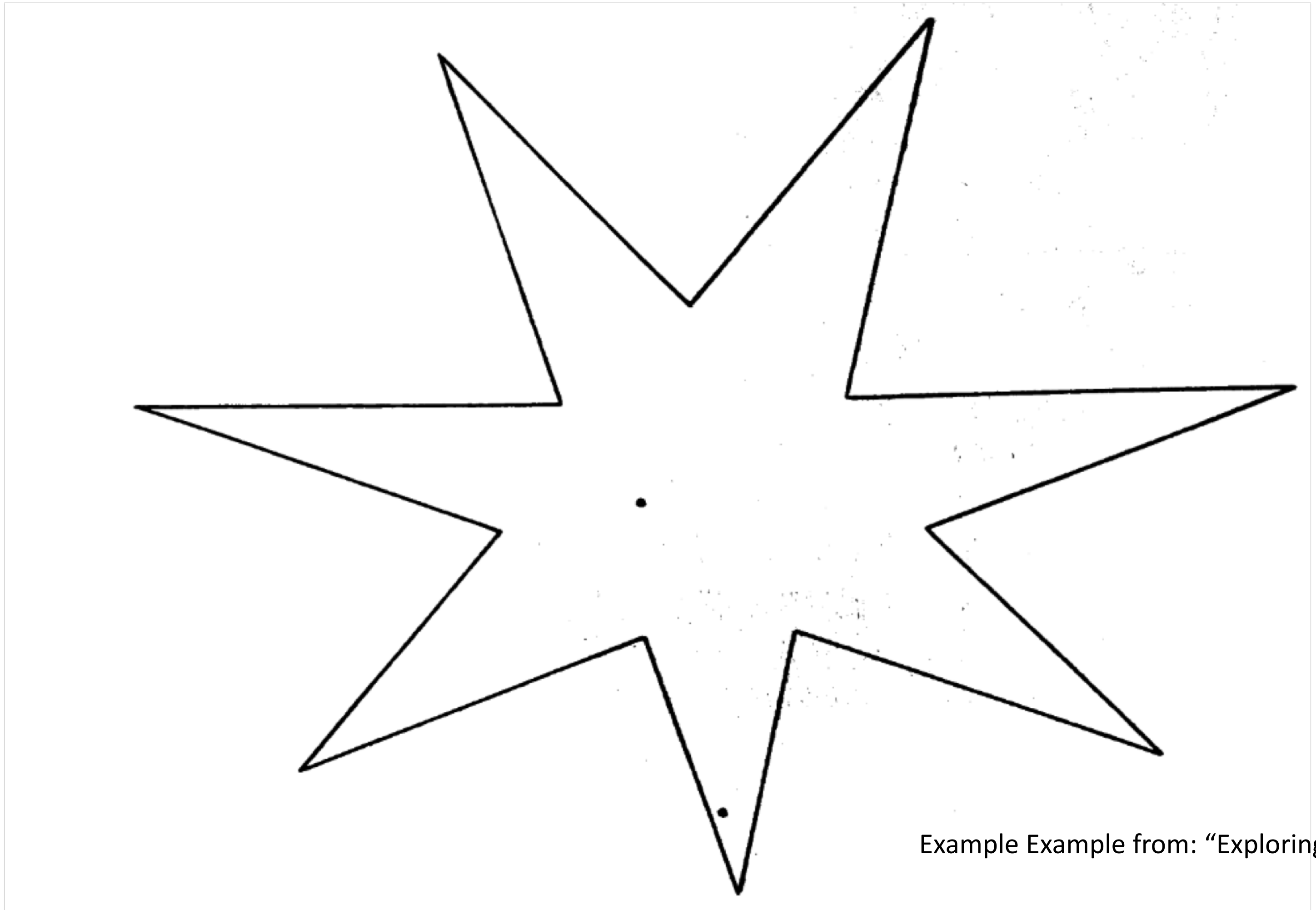
NOTES

- We should cover problem statements more explicitly…
- Requirements and problem statemens could be considered synonyms…how are they different here?
- Should recommend that problem statements often include a motivation:

  “will reduce by costs by $XXXX”.

Lecture 3

Ambiguity in Problem Statements and High-Level Requirements



Example Example from: “Exploring requirements – quality before design” – Weinberg and Gause

Agenda

Guest Presentation:

Skot Wiedmann

Review:

Quad Charts
Ambiguity

Lecture:

Problem Statements

Slides today are based mostly and often directly on Gause and Weinberg.

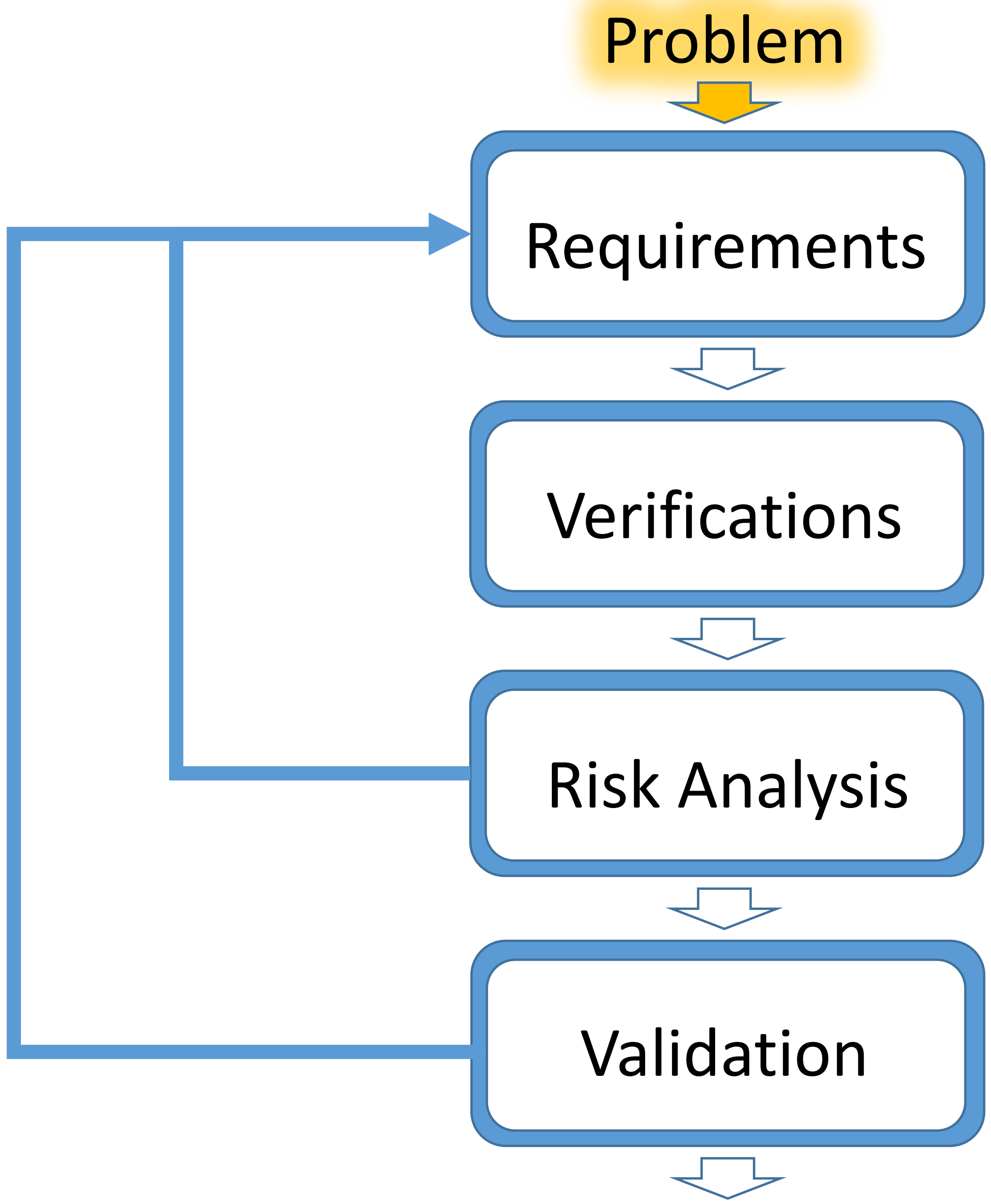
Guest Lecture

Skot Wiedmann

- Bachelor’s Degree in Digital Interactivity from the University of Wisconsin
- Master’s in Fine Arts in New Media from UIUC
- Entrepreneur who has started multiple companies
- Lecturer in the Fine Arts Department bringing together electronics and media
- Works in ECE in the electronic services shop
- More circuit knowledge than just about anyone in this building.

Reviewing our problem statements

- A few notes
  - WE - When I talk about documentation around projects we are talking about/working on, I try to use the language of “we”. “We” did this well, “we” could do better in this area. The reason I do this is that when working as a team, we succeed together and we fail together.
  - CONSTRUCTIVE CRITICISM - Good design requires honesty, courtesy, and respect of your colleagues. The recognition of flaws in a document, design, or product is a critical part of the design process. What matters is how you present that criticism to the team and the team’s willingness to recognize the value of that criticism.
  - PERSPECTIVE - Everyone should have the opportunity/be encouraged to speak. If certain voices are over or underemphasized, you are losing out on perspective.
- OK….so let’s talk about problem statements.

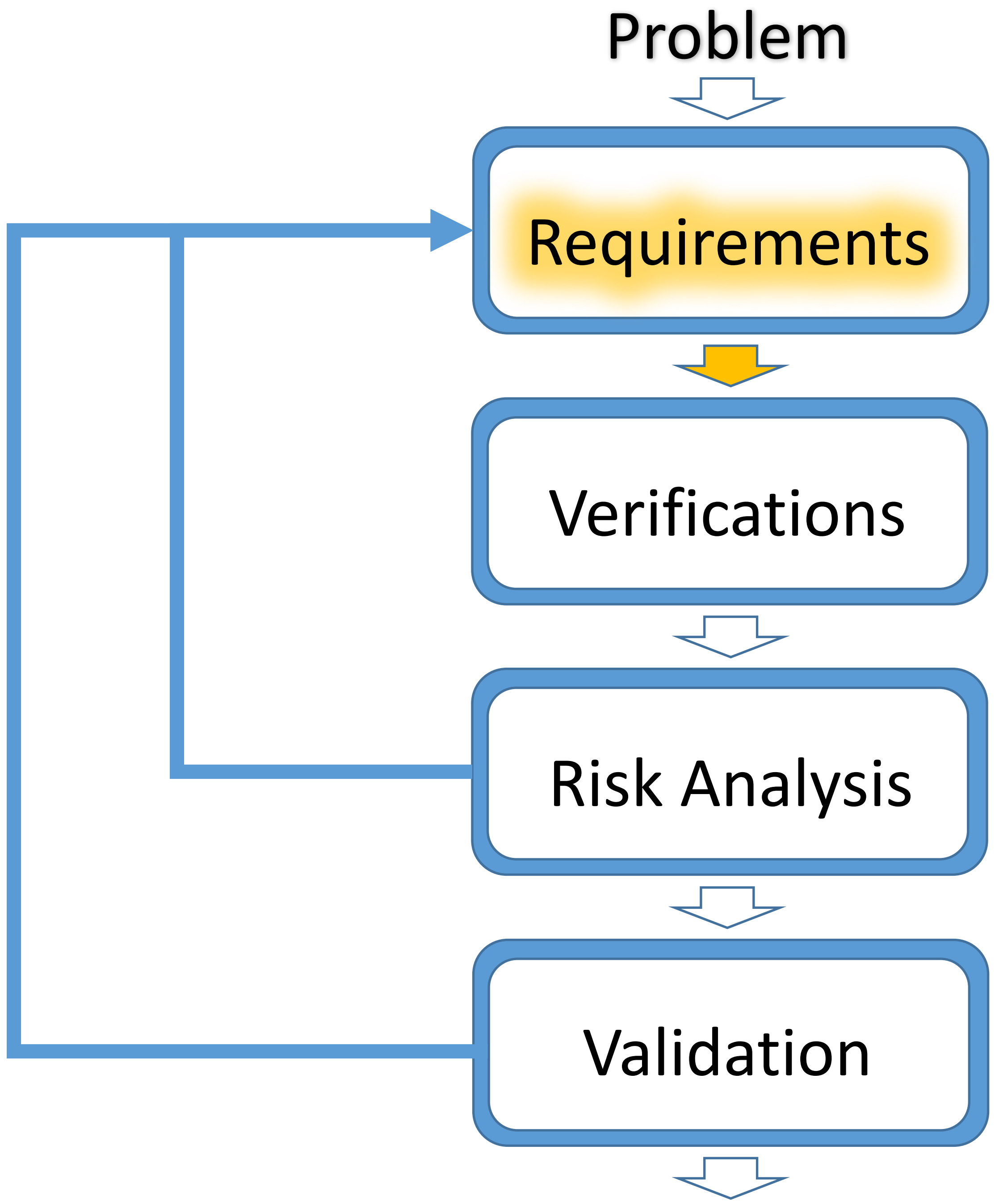


Problem Statement:

A description of what you are trying to solve and why.

A problem statement should be:

- Brief
- Clear
- Unambiguous



Requirements:

A set of statements that describe the attributes your system must have to solve your problem.

They should be:

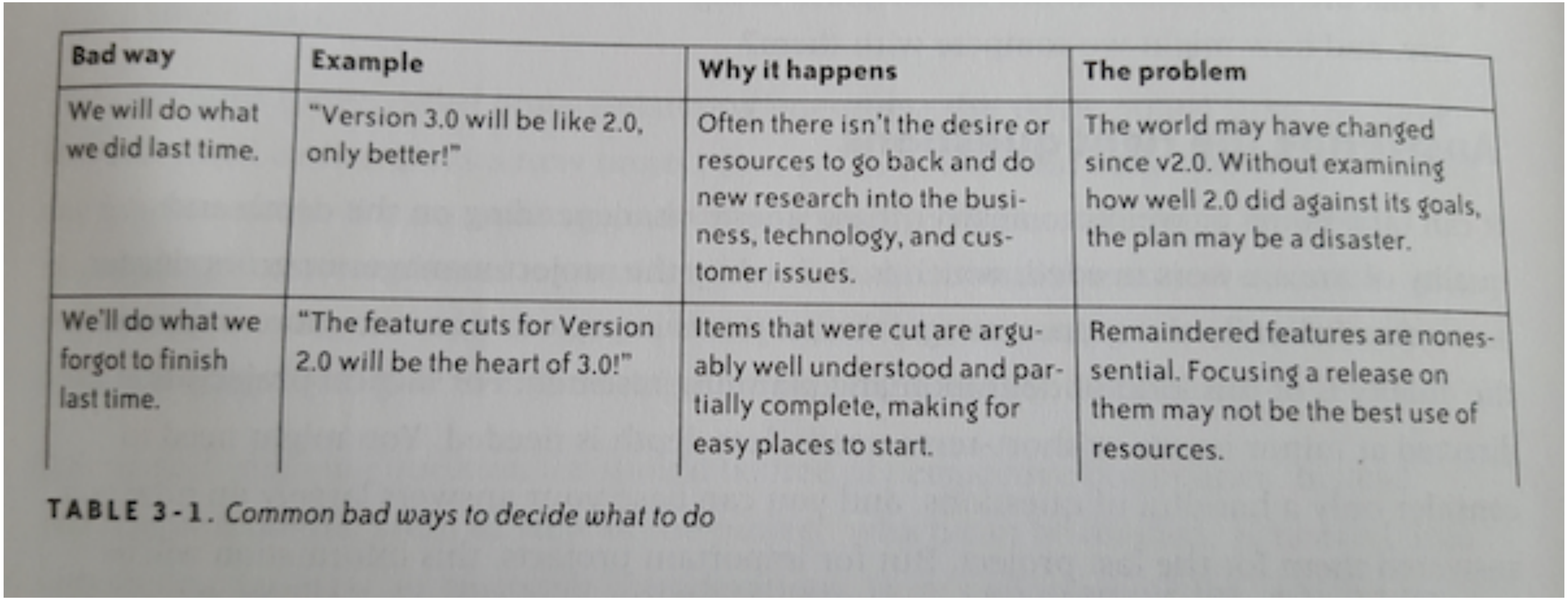
- Quantifiable
- Relevant
- Detailed

Today we are concentrating on ambiguity…

- Why?

Google search and Youtube

Examples of ambiguity in project statements.

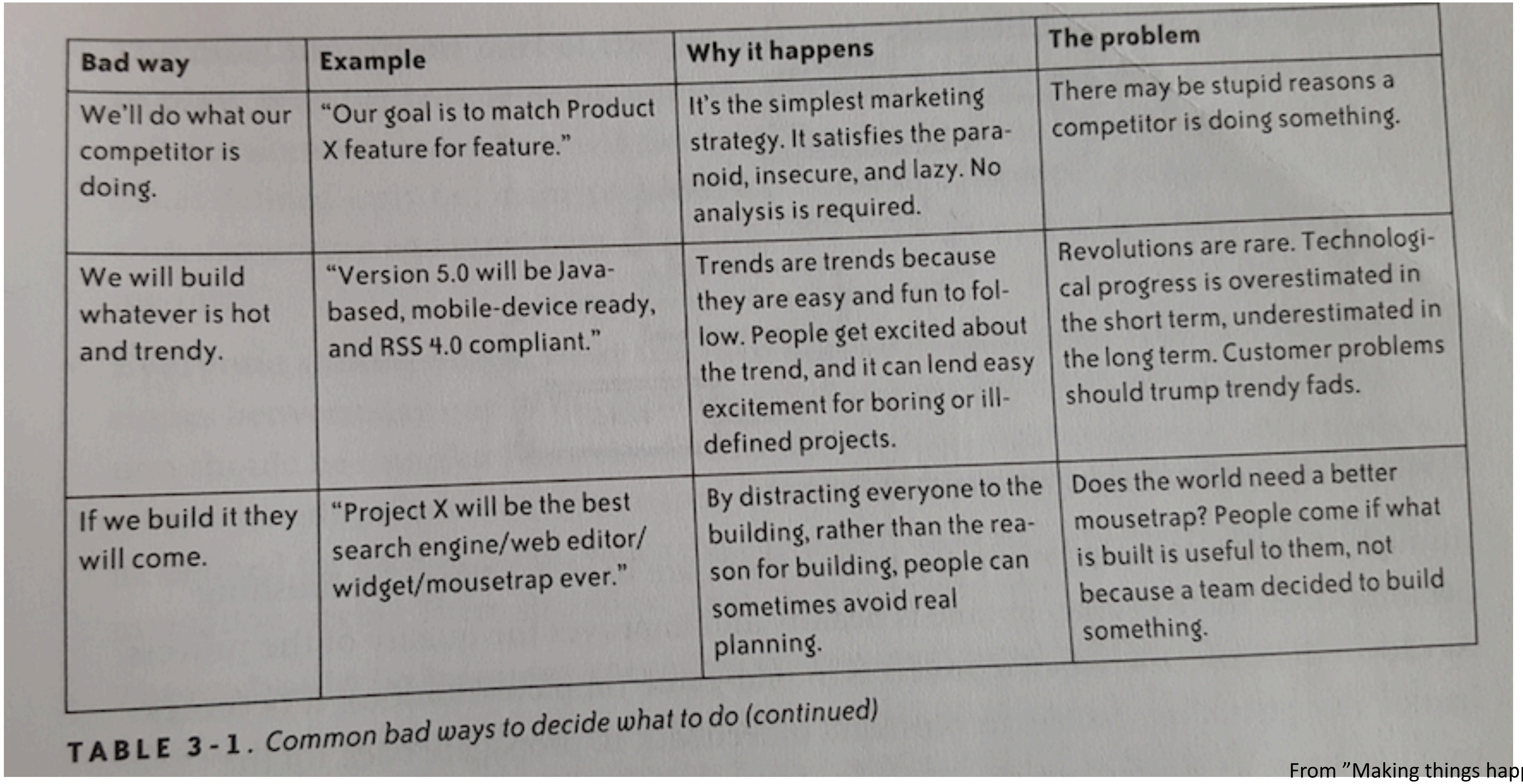


Bad way	Example	Why it happens	The problem
We will do what we did last time.	"Version 3.0 will be like 2.0, only better!"	Often there isn't the desire or resources to go back and do new research into the business, technology, and customer issues.	The world may have changed since v2.0. Without examining how well 2.0 did against its goals, the plan may be a disaster.
We'll do what we forgot to finish last time.	"The feature cuts for Version 2.0 will be the heart of 3.0!"	Items that were cut are arguably well understood and partially complete, making for easy places to start.	Remaindered features are nonessential. Focusing a release on them may not be the best use of resources.

TABLE 3-1. *Common bad ways to decide what to do*

From "Making things happen" Berkun

Examples of ambiguity in project statements.



Bad way	Example	Why it happens	The problem
We'll do what our competitor is doing.	"Our goal is to match Product X feature for feature."	It's the simplest marketing strategy. It satisfies the paranoid, insecure, and lazy. No analysis is required.	There may be stupid reasons a competitor is doing something.
We will build whatever is hot and trendy.	"Version 5.0 will be Java-based, mobile-device ready, and RSS 4.0 compliant."	Trends are trends because they are easy and fun to follow. People get excited about the trend, and it can lend easy excitement for boring or ill-defined projects.	Revolutions are rare. Technological progress is overestimated in the short term, underestimated in the long term. Customer problems should trump trendy fads.
If we build it they will come.	"Project X will be the best search engine/web editor/widget/mousetrap ever."	By distracting everyone to the building, rather than the reason for building, people can sometimes avoid real planning.	Does the world need a better mousetrap? People come if what is built is useful to them, not because a team decided to build something.

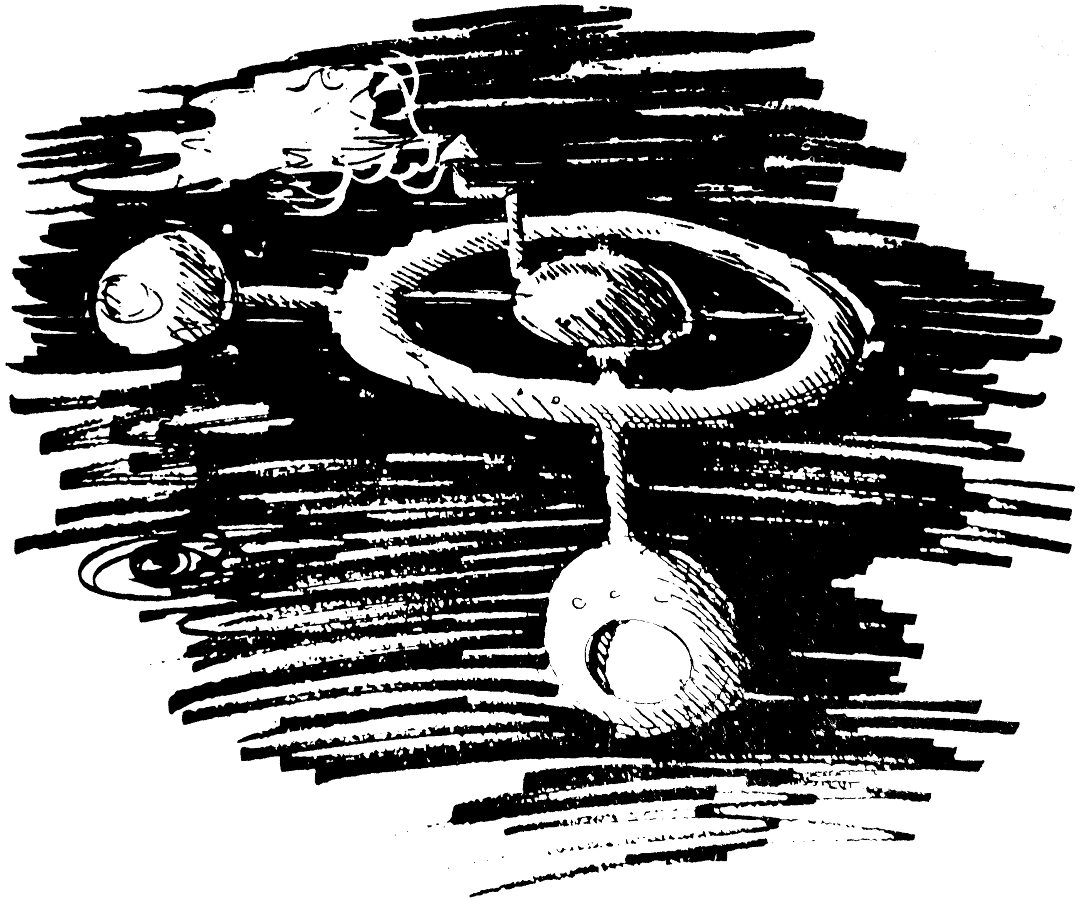
TABLE 3-1. *Common bad ways to decide what to do (continued)*

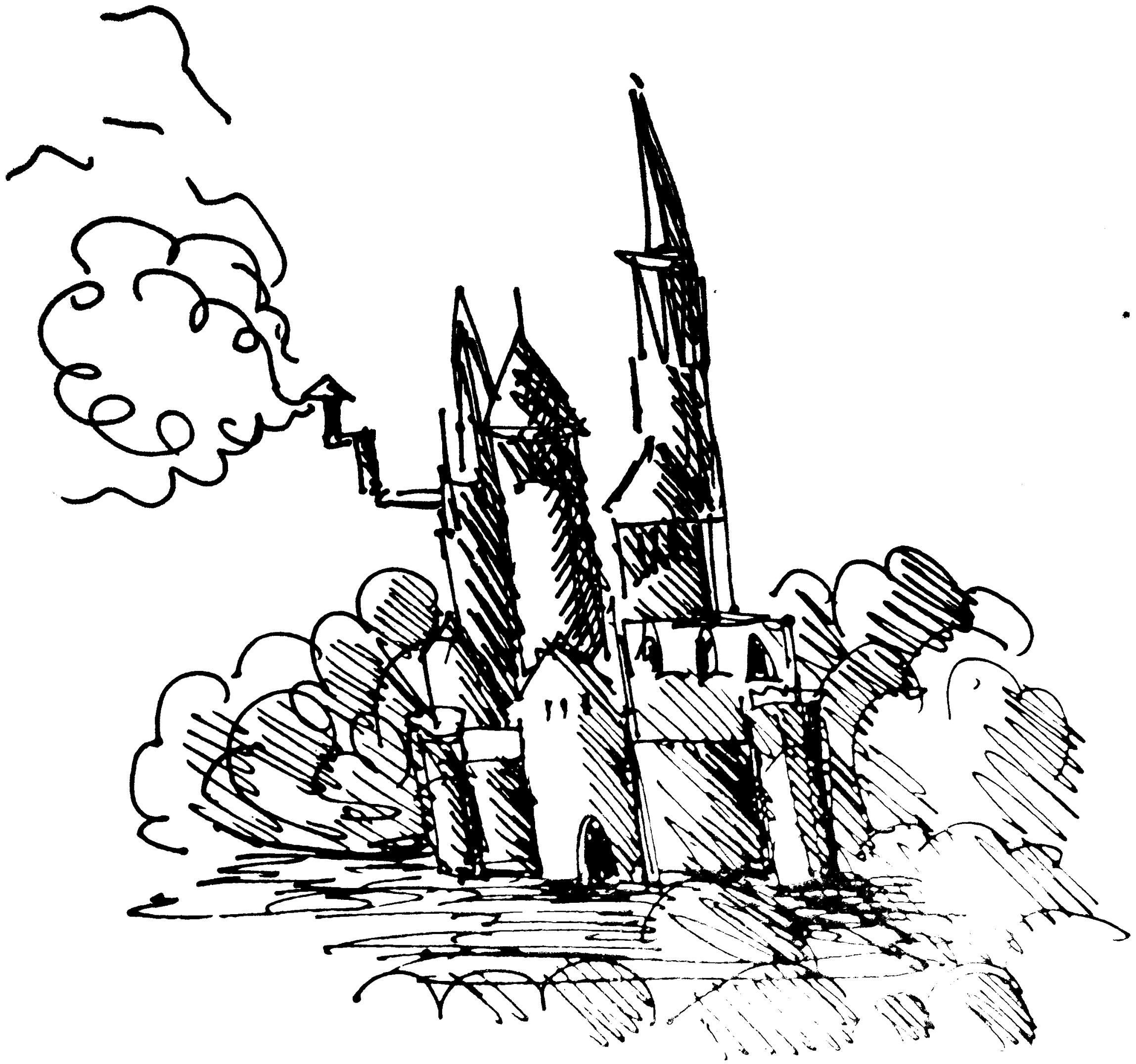
From "Making things happen" Berkun

Why are we starting like this?

Create a means to protect a small group of humans from the hostile environment.

Example from: “Exploring requirements – quality before design” – Weinberg and Gause





Example from: “Exploring requirements – quality before design”

- Stating the obvious….

“If you do not know what you want,
you are quite unlikely to get it.”

- In engineering design, the description of what you want is captured by the problem statement and “high-level” requirements.
- We call them “high-level” to differentiate them from lower level specifications.

Example:

- Requirement - The car must be safer than the current model.
- Specification - The airbags must deploy in under 0.3 seconds.

Example from: “Exploring requirements – quality before design”

- Stating the obvious….

“If you do not know what you want,
you are quite unlikely to get it.”

- In engineering design, the description of what you want is captured by the problem statement and “high-level” requirements.
- We call them “high-level” to differentiate them from lower level specifications.

  Example:
  - Requirement - The car must be safer than the current model.
  - Specification - The airbags must deploy in under 0.3 seconds.

Ambiguity - the source of all that is evil…..

- Missing requirements
  - Examples?

- Ambiguous language
  - Words
    - Small, large, lots, many, very, group

- Introduced elements
  - Notice our definition never actually said structure…we made that assumption ourselves.

Example from: “Exploring requirements – quality before design” – Weinberg and Gause

Humans…another source of ambiguity…

- Let’s take a detour….

INVISIBLE GORILLA

Certainly, this is not a problem for all of us?

- Piece of paper, write down.
- How many points were in the star on the title slide of this presentation?

Certainly, this is not a problem for all of us?

- Active discussion.
- How many points were in the star on the title slide of this presentation?
  - 1 class answer.

Answers from Gause and Weinberg

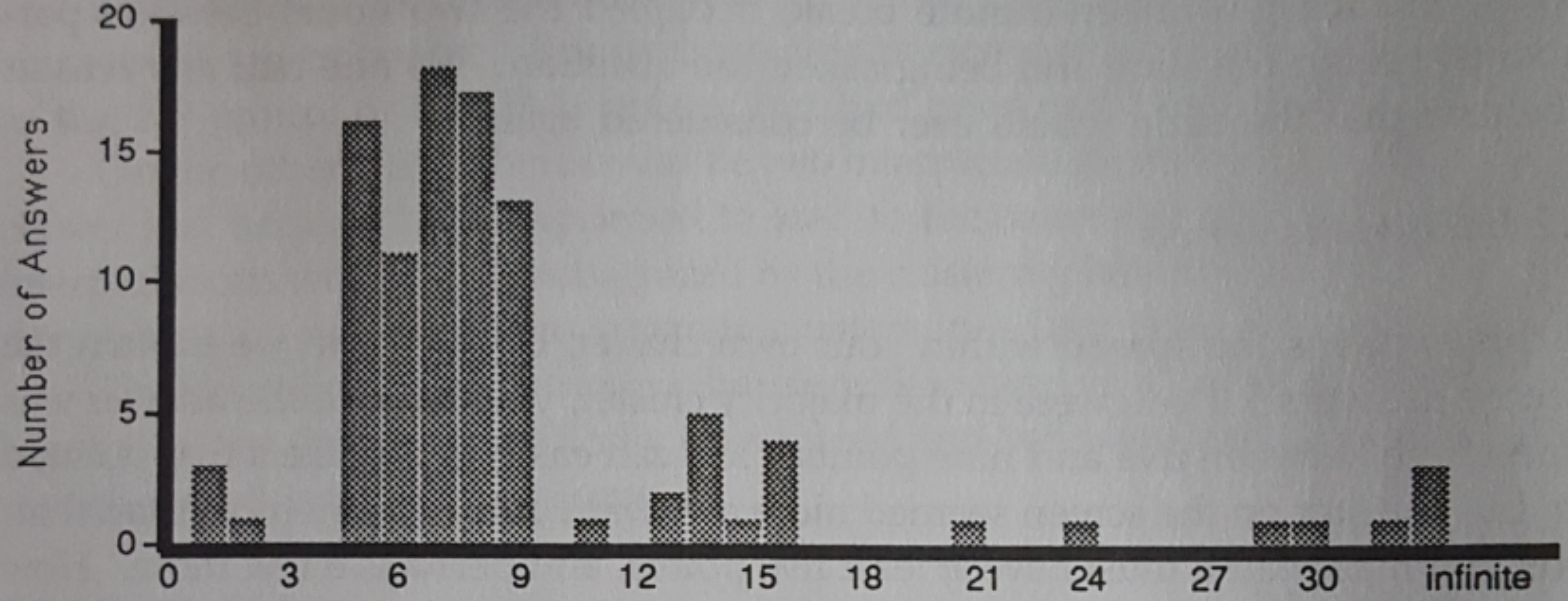


Figure 3-3. Histogram of answers to the "how many points" question.

Example from: "Exploring requirements – quality before design" – Weinberg and Gause

Answers from Gause and Weinberg

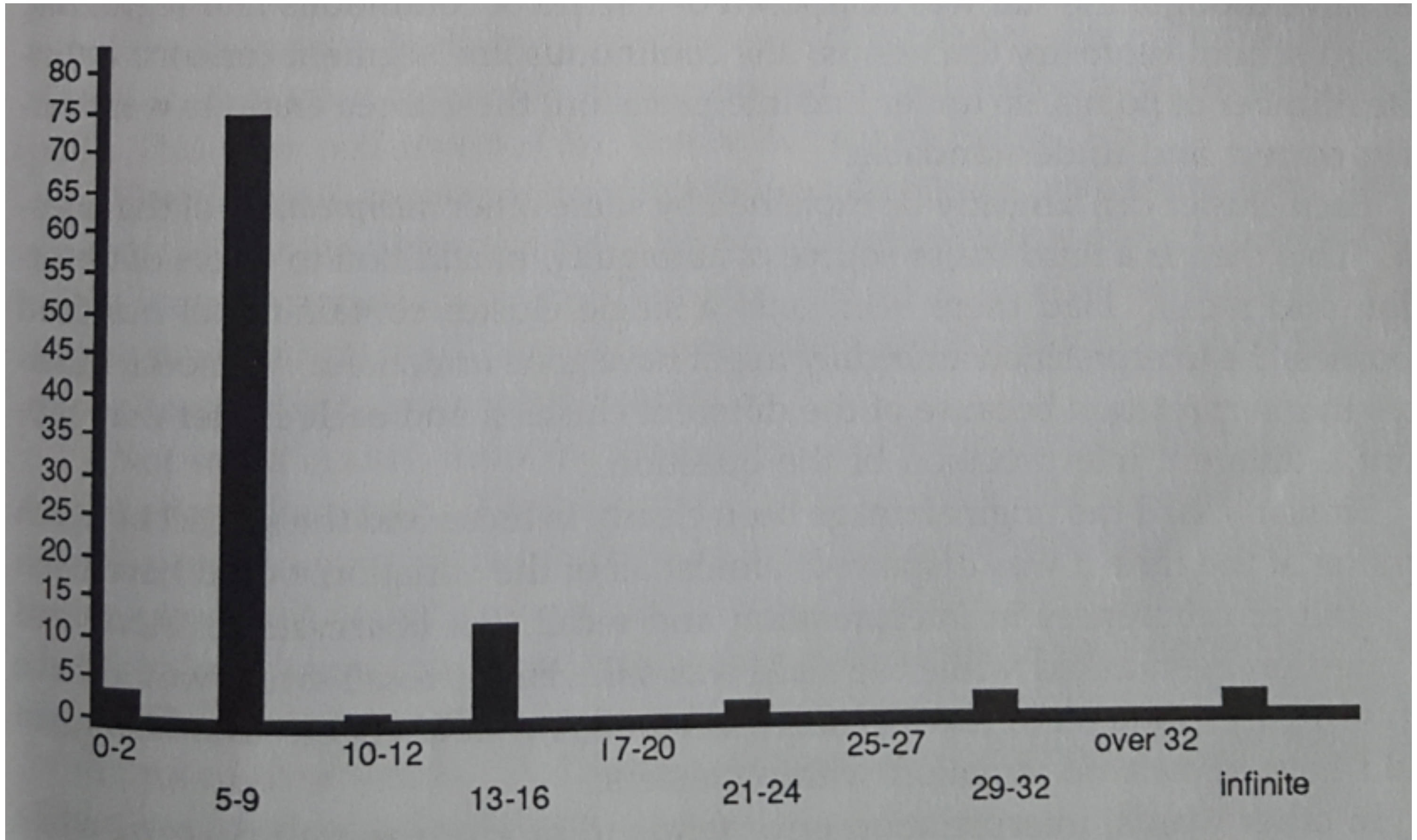


Figure 3-4. Histogram of clumped answers to the question.

Example from: “Exploring requirements – quality before design” – Weinberg and Gause

Humans…another source of ambiguity…

- Observational errors
  - We are predisposed to ignore information that we do not think is important.
- Recall errors
  - We are bombarded with information every day, our memories of this information (recall) is often wrong.
- Interpretation errors
- Hybrid errors
  - Mixtures of these types of errors

Example from: “Exploring requirements – quality before design” – Weinberg and Gause

Answers from Gause and Weinberg (100 people at conference)

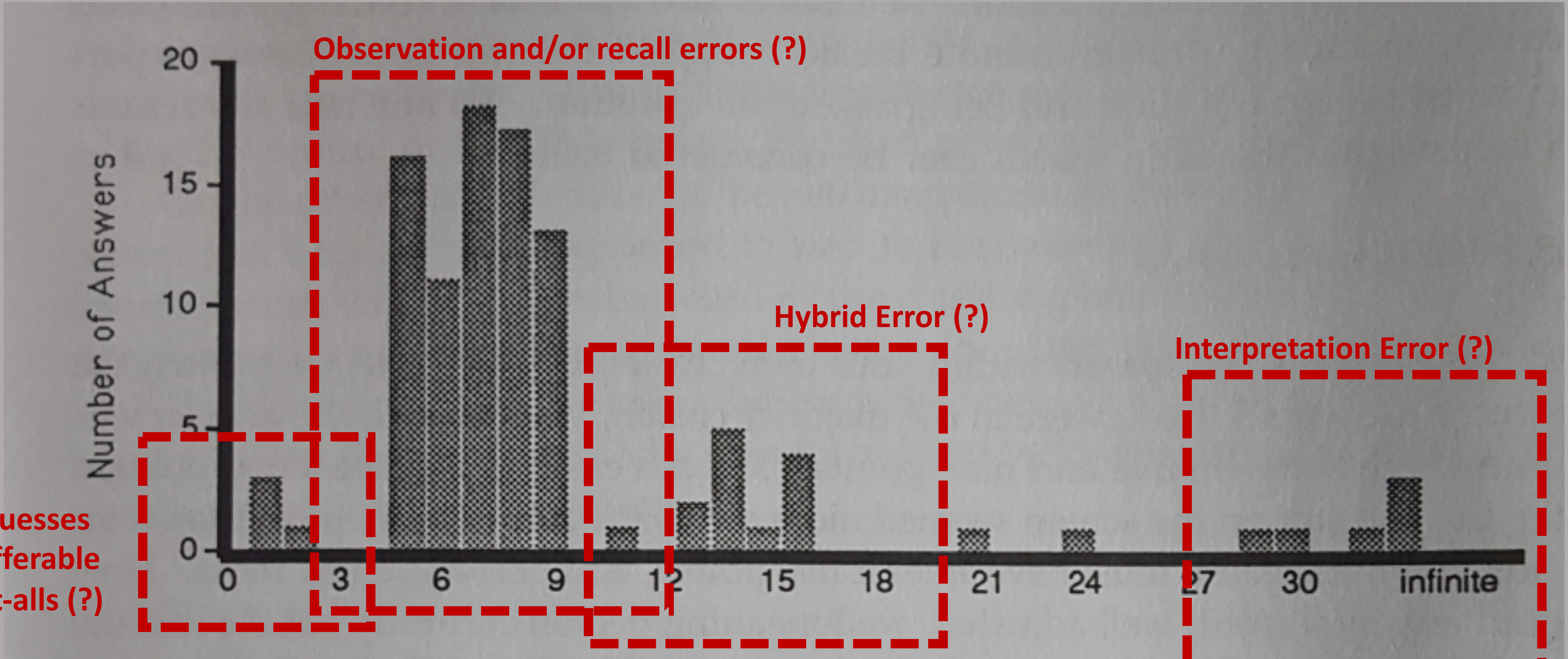


Figure 3-3. Histogram of answers to the "how many points" question.

Example from: "Exploring requirements – quality before design" – Weinberg and Gause

Another source of ambiguity

- Write down the question that I just asked you.

Another source of ambiguity

- Write down the question that I just asked you.
- Not having the problem statement/requirements visible can introduce ambiguity...

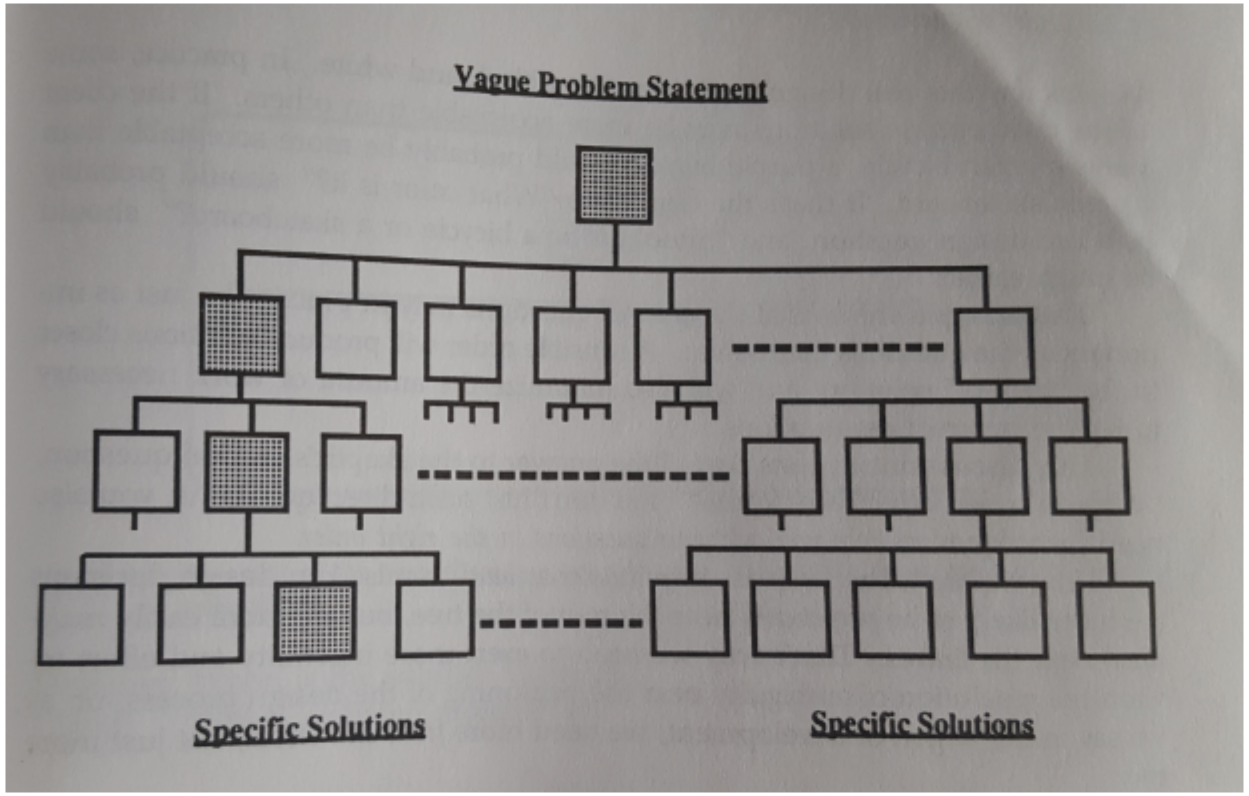
Methods for Combating Ambiguity?

- Open discussion
  - Was our class answer better than the individual answers?
  - Why?
- Direct questions
  - We can envision our design space as a decision tree, the problem statement as the root of that decision tree, and the requirements as a set of decisions to help us navigate that tree.
  - By asking direct questions, we can intuitively navigate that tree.
- Starting points
- Context free questions
- Considering the customer and user
- Brainstorming

Example Example from: “Exploring requirements – quality before design” – Weinberg and Gause

Some issues with this approach?

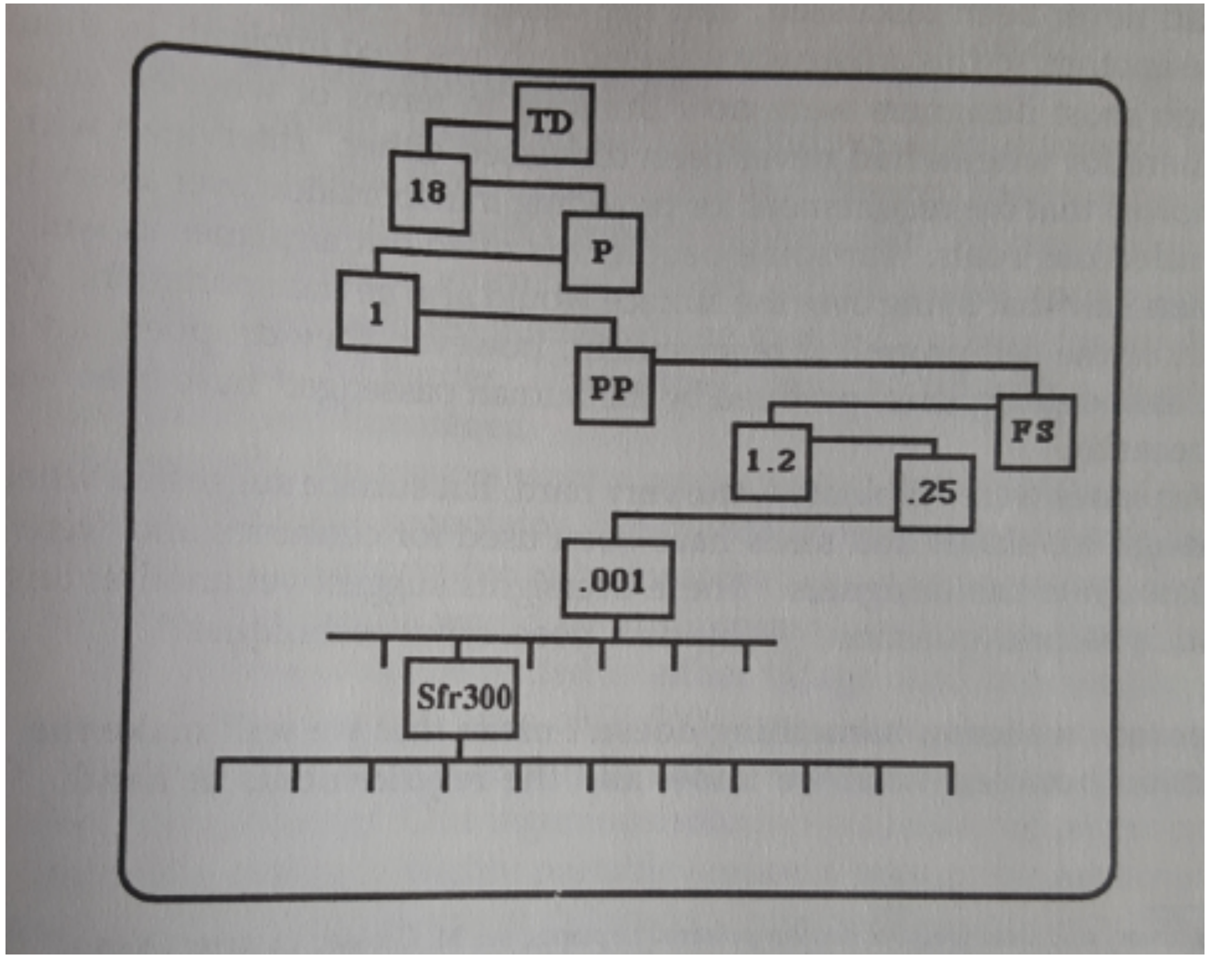
- Open discussion
  - Was our class answer better than the individual answers?
  - Why?
- Direct questions
  - We can envision our design space as a decision tree, the problem statement as the root of that decision tree, and the requirements as a set of decisions to help us navigate that tree.
  - By asking direct questions, we can intuitively navigate that tree.



Example Example from: "Exploring requirements – quality before design" – Weinberg and Gause

Swiss Client

- Design a transportation device.
  - By when? 18 months
  - What is to be transported? people
  - How many people at a time? 1
  - How is the device powered? Natural forces
  - What kind of surface? A hard flat surface
  - How far is it transporting people? 1.2 miles
  - How fast must it be? 0.25 mile per hour
  - How reliable must it be? 1 failure every 1000 hours
  - Cost limitations? $300



Example Example from: “Exploring requirements – quality before design” – Weinberg and Gause

Swiss Client

https://www.walmart.com/ip/20-Durban-Bikes-Rio-Up-Folding-Bike-Turquoise/55090334?wmlspartner=wlpa&selectedSellerId=0&adid=22222222227064416517&wl0=&wl1=s&wl2=c&wl3=167044087581&wl4=pla-274170482069&wl5=9022185&wl6=&wl7=&wl8=&wl9=pla&wl10=8175035&wl11=online&wl12=55090334&wl13=&veh=sem

https://www.google.com/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=0ahUKEwih4rac5uzRAhXI6oMKHQFyBTsQjRwIBw&url=http%3A%2F%2Fwww.brooks-range.com%2FAll-In-One-Rescue-Sled-Tarp-p%2F1101-100-0.htm&psig=AFQjCNGwUFWP6E-LQj5bOj9s2QKHlyxk3Q&ust=1485966337454973

This is an odd example…

- The point, much like our gorilla, is that we are rarely aware of what we have overlooked.
- Decision tree can be helpful, but quickly becomes too large.
- Can reduce ambiguity by sharing results of decision tree with client.
- A last note, make sure to give your customer what they want instead of what you THINK they need. If you can't convince them you are right, either make what they want or opt out of the project.

Example Example from: "Exploring requirements – quality before design" – Weinberg and Gause

Starting with a problem

- To reduce ambiguity, we need some starting points.
- Every design can be viewed as an attempt to solve a problem:
  - "A difference between things as perceived and things as desired"
- There are different ways, however, to start moving towards a problem.

Example Example from: "Exploring requirements – quality before design" – Weinberg and Gause

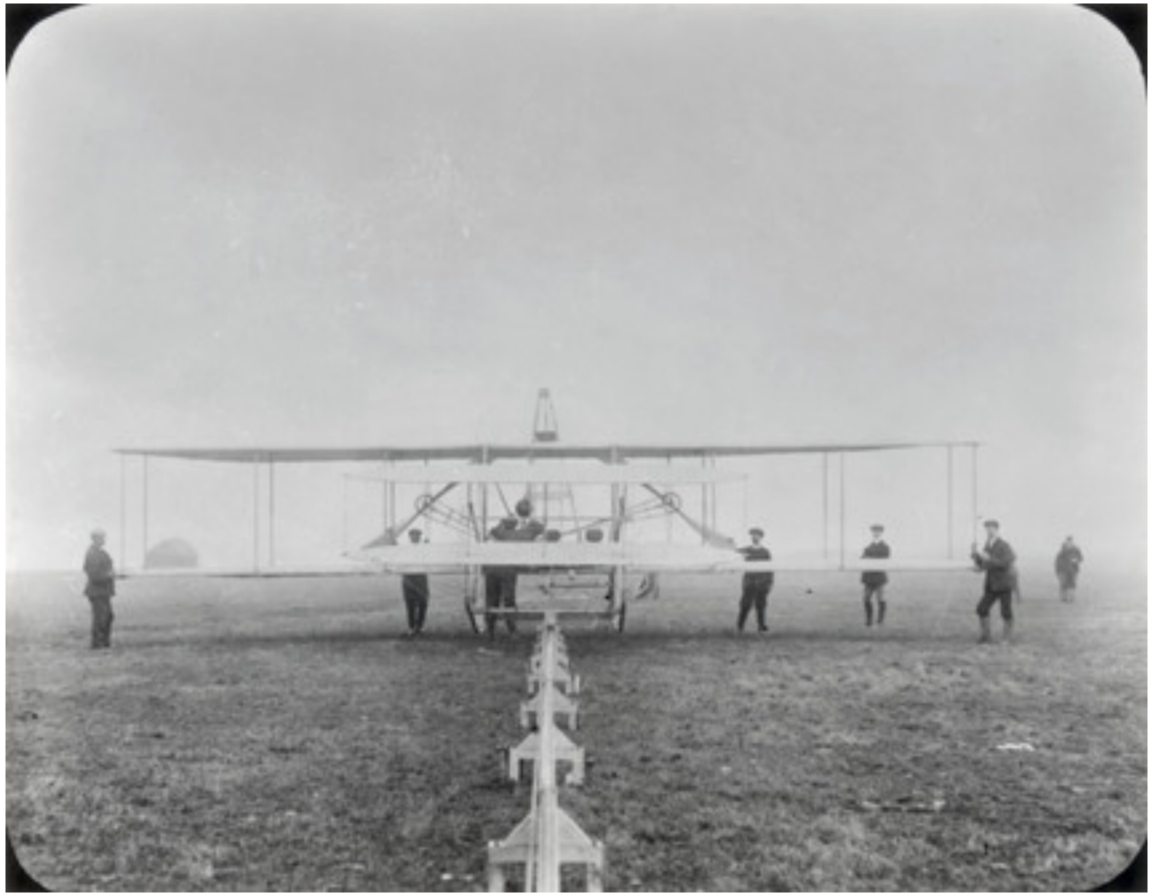
Six starting points that aren't problems, but can made into them…

- Solution idea
  - Starting your design with a solution rather than a problem
    - Example: "We need to attract more students"
      - Why?
        - To fill the dorms?
        - To expand the campus?
        - To increase our rejection rate?
      - What kind of students?
      - How many is more?
- Technology idea
  - Have a new technology in need of a problem
    - Post-It note was a failed attempt to create a new kind of adhesive
  - Problem statements of these kinds can be thought of as creating a desire for your technology.

Example Example from: "Exploring requirements – quality before design" – Weinberg and Gause

Six starting points that aren't problems…

- Simile
  - Build a product like XXXX.
    - What does like mean?
  - Can be helpful during ideation, but must be defined to form a real problem.
- Norm
  - The bane of ECE445…"smart – XXX"
  - What is smart about it?
  - Constrains your thinking
  - Similarly to similes, can be useful during ideation
- Mockup



https://www.google.com/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=0ahUKEwjx7ui3qu3RAhVLyoMKHdtwAcAQjRwIBw&url=http%3A%2F%2Fwww.ssplprints.com%2Fimage%2F85233%2Fwright-brothers-type-a-aeroplane-on-its-launching-rail-sheppey-kent-c-1910&bvm=bv.145822982,d.cGw&psig=AFQjCNG8ViMNLjOvAxtBbdEwl26bIrTn7A&ust=1485984670708800

Example Example from: "Exploring requirements – quality before design" – Weinberg and Gause

Different starting points

- It can be helpful to start your requirements/ideation process over from multiple starting points.
- Example: If you tried a norm, then try a mockup.

Example Example from: “Exploring requirements – quality before design” – Weinberg and Gause

Context Free Questions

- A set of questions that can be asked regardless of the design project.

- Context Free Process Questions
  - Who is the client for the XXX project?
  - What is a highly successful solution really worth to this client?
  - What is the real reason for wanting to solve this problem?
  - Should we use a single team? More than one? Who should be on those teams?
  - How much time do we have?
  - Where else can the solution to this problem be obtained? Can we copy an existing solution?

Example Example from: “Exploring requirements – quality before design” – Weinberg and Gause

Context Free Questions

- A set of questions that can be asked regardless of the design project.

- Context Free Process Questions
  - Who is the client for the XXX project?
  - What is a highly successful solution really worth to this client?
  - What is the real reason for wanting to solve this problem?
  - Should we use a single team? More than one? Who should be on those teams?
  - How much time do we have?
  - Where else can the solution to this problem be obtained? Can we copy an existing solution?

  - Cowboy example

Example Example from: “Exploring requirements – quality before design” – Weinberg and Gause

Context Free Product Questions

- What problems does this system solve?
- What problems could this system create?
- What environment is this system likely to encounter?
- What kind of precision is required?

Example Example from: “Exploring requirements – quality before design” – Weinberg and Gause

Metaquestions – questions about questions

- Do my questions seem relevant?
- Are you the right person to answer this question?
- May I write your answers down and read them back to you?
- Is there someplace we can see the environment where this product will be used?
- Is there anything else I should be asking you?
- Is there anything you would like to ask me?
- May I ask you more questions later?

Example Example from: “Exploring requirements – quality before design” – Weinberg and Gause

Advantages

- Can have a list of questions to answer in advance.
- They can help with social awkwardness if working with an external customer.
- Help you focus on high level issues.
- Help raise issues that may have been implicit.
- Can help to elucidate competing assumptions.

Example Example from: “Exploring requirements – quality before design” – Weinberg and Gause

Getting the right people involved

- Often, and very often in 445, we design something without the MOST important people in mind….the customers and the users.
  - Customer (or client)
    - The people pay you for your product
    - “Who is the customer” should always be your first context-free question.
  - User - anyone who is affected by the project
    - Customers are users…
    - Users are not always the customers
      - Why wouldn’t a line of toys that children love sell?

Example Example from: “Exploring requirements – quality before design” – Weinberg and Gause

Why include the user?

- Customers pay for the product
- Users make or break products, many tools are never used, without use…there will be no repeat customers…
- Users should be involved in the design process as well

Example Example from: “Exploring requirements – quality before design” – Weinberg and Gause

The railroad paradox

1. Product is not satisfactory.
2. Because of 1, no one uses the product.
3. Potential users ask for a better product.
4. Because of 2, request is denied.

Products can create new users…

- Speed limits and traffic fatalities

Example Example from: “Exploring requirements – quality before design” – Weinberg and Gause

Optimizing a product

- May be possible to improve a product for some without making it worse for others…Pareto optimization.
- Others (Veblen) argue that making a product better for one user always makes it worse for some others.
- When we change something in a design, we should consider both groups.

Example Example from: “Exploring requirements – quality before design” – Weinberg and Gause

Considering important users

- Identifying users and whether they should be dealt with is important.
- Start with a huge list
- Prune list
- Then, for a design we can consider whether the it should be friendly to them, ignore them, or be unfriendly to them
  - Unfriendly example, medicine bottles for children

Example Example from: “Exploring requirements – quality before design” – Weinberg and Gause

Participation

- Who should be involved?
  - Ideally, every potential user
  - Sometimes this is possible
  - Often is not…if not, take a sample?
  - Don’t confuse a sample with being “everyone”.
- When should they be involved?
  - Sometimes they are members of the team.
    - Can be useful if there is a small user group known ahead of time.
  - Part time user involvement more common
    - Have a plan…may drift to 0.

Example Example from: “Exploring requirements – quality before design” – Weinberg and Gause

Participation

- How to get judgements?
  - Mockups
  - Models
  - Be careful to differentiate from opinions and data from experiments
  - Make note of whether your representations of the product are representative of the real world.
    - Example: tops of buildings…

- Final note: when it comes to including customers and users, have a plan and make it public.

Example Example from: “Exploring requirements – quality before design” – Weinberg and Gause

Participation

- How to get judgements?
  - Mockups
  - Models
  - Be careful to differentiate from opinions and data from experiments
  - Make note of whether your representations of the product are representative of the real world.
    - Example: tops of buildings…

- Final note: when it comes to including customers and users, have a plan and make it public.

Example Example from: “Exploring requirements – quality before design” – Weinberg and Gause

Ambiguity Reduction Heuristics

- Memorization
  - Note: According to Gause and Weinberg…programmers can find bugs easier if they try to memorize code. Areas that are harder to memorize more likely to be in error.
  - If a problem statement or requirement are clear, they may be easier to remember.
  - Try having the team memorize them, analyze errors to determine what is unclear.

- Poll people
  - How many points were in the star?
  - After time and discussion repoll.

Example Example from: “Exploring requirements – quality before design” – Weinberg and Gause

Mary had a little lamb

- Analyze things word for word and then in combinations of words
  - Meaning may not be clear or can be lost over time
    - Half a pound of tupenny rice,
      Half a pound of treacle.
      That's the way the money goes,
      Pop! goes the weasel.
  - What does "Mary had a little lamb" tell us?
    - *Mary* had a little lamb
    - Mary *had* a little lamb
    - *Mary had* a little lamb

Example Example from: "Exploring requirements – quality before design" – Weinberg and Gause

Mary conned the trader

- What if we were to substitute synonyms for the words in the problem statement?

"Mary had a little lamb"

- Had
  - Possess own hold
  - To obtain, accept
  - To partake
  - Trick
- Lamb
  - Young sheep, less than 1 year of age
  - Young of various animals
  - Gentle or weak person
  - Easily cheated or deceived, especially in securities

Examples

- Mary owned a young antelope
- Mary had a little bit of lamp for dinner
- Mary tricked a little sheep under one year of age
- Mary had a small gentle person
- Mary bribed a small person trading in securities who was easily cheated

Example Example from: "Exploring requirements – quality before design" – Weinberg and Gause

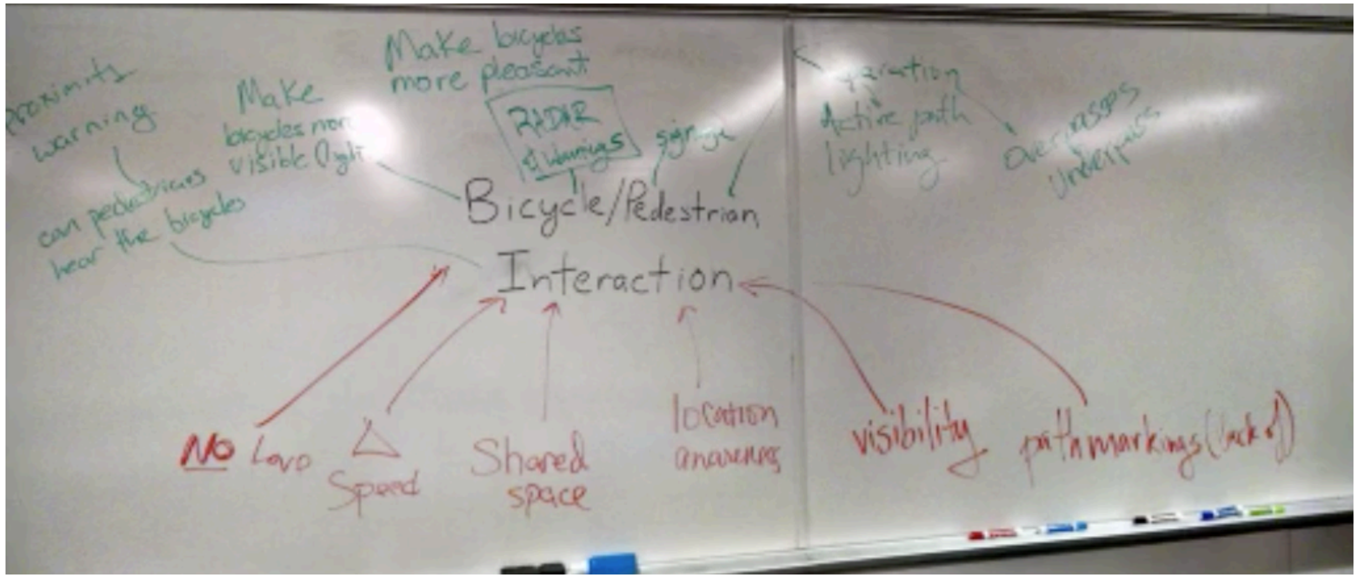
Some specific brainstorming methods

- Generating more ideas
  - Idea tree
- Reducing the number of ideas
  - Voting
- Exploring an idea further
  - Diving deeper
  - Reverse brainstorming

Example Example from: “Exploring requirements – quality before design” – Weinberg and Gause

The idea tree

- The high-level problem forms the trunk
- Try to identify the roots of the problem
- Be inspired by the roots to grow solution branches



Example Example from: "Exploring requirements – quality before design" – Weinberg and Gause

How reverse brainstorming works

- Come up with ideas as to how you could cause this problem or how you could make the product worse

- Reverse these ideas of how to make things worse into potential ways to solve the problem or make the product better

How voting works

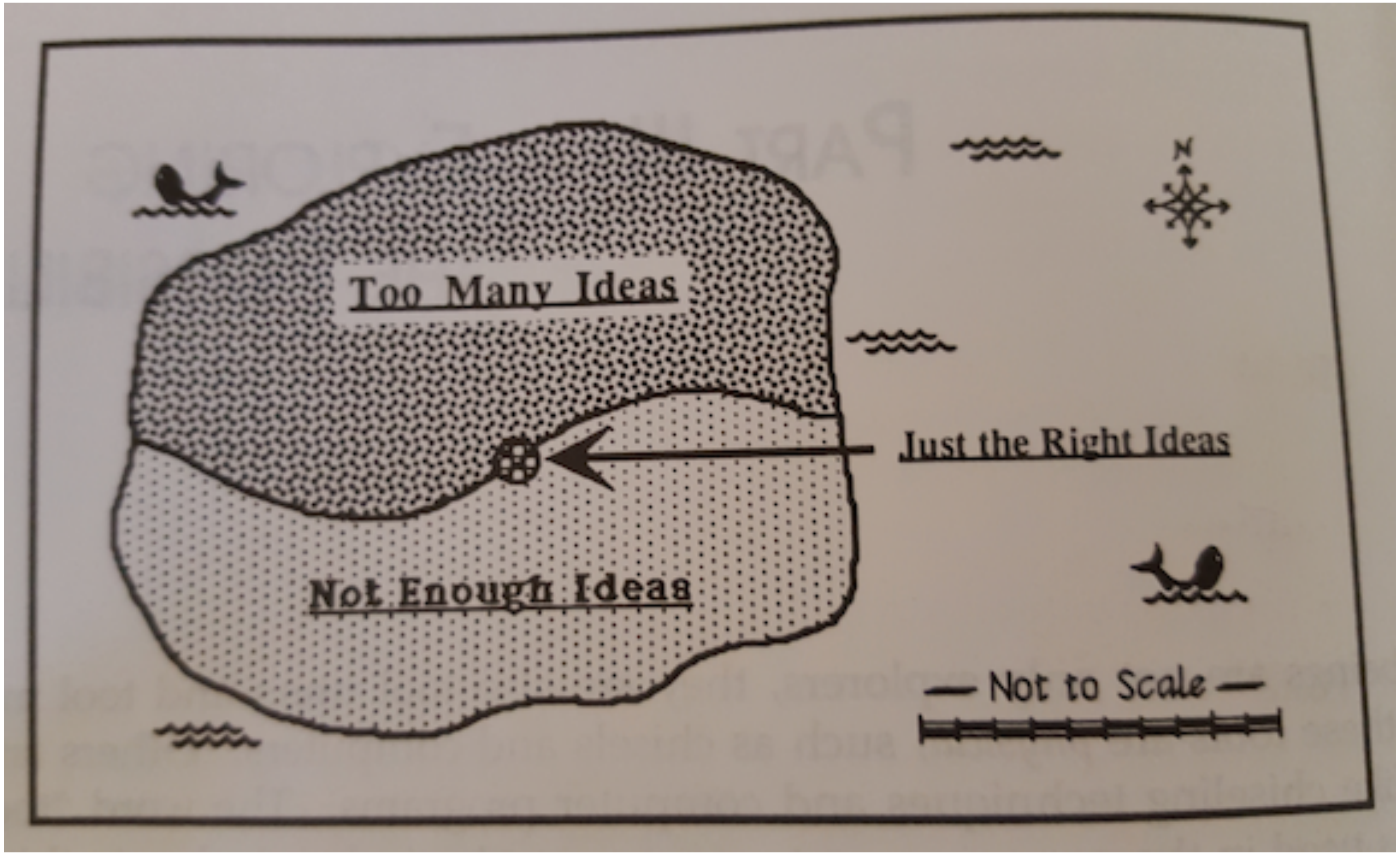
- Deselect at end
  - 2 positive votes
  - 1 negative vote

Diving deeper into an idea

Post it notes can be used to clarify idea potential:

Yellow: Core idea
Blue: improvement or modification
Red: problem or challenge

The "jungle of ideas"


Too Many Ideas
Just the Right Ideas
Not Enough Ideas
N
Not to Scale