GREAT SOFTWARE DEVELOPMENT - I

Content from Chapter 1 of "Head First Software Development", Pilone et al.

Miami University Software Technology & Analysis Group (MUSTANG)
Computer Science & Software Engineering
Miami University, Oxford, Ohio, USA

AGENDA

- Free Advice
- Review
- Happy Customer
- Iteration
- High-Level Requirements (features)
- Iteration Plan

AGENDA

- Free Advice
- Review
- Happy Customer
- Iteration
- High-Level Requirements (features)
- Iteration Plan

GET HELP!

- Disabilities
  - If you need accommodations, please let me know

- 10% of Miami students have a learning disability
- 60% of Miami students report alcohol use
- 89% of Miami students report feeling "overwhelmed"
- 20% of Miami students have a diagnosed depressive disorder
- 14 % of Miami students have a diagnosed anxiety disorder

GET HELP!

- You are not alone
- Get help
  - Come see me!
- Find healthy ways to manage

AGENDA

- Free Advice
- Review
- Happy Customer
- Iteration
- High-Level Requirements (features)
- Iteration Plan

AGENDA

- Free Advice
- Review
- Happy Customer
- Iteration
- High-Level Requirements (features)
- Iteration Plan

HAPPY CUSTOMER

- "If your customer is unhappy, everyone is unhappy!" [TEXTBOOK]

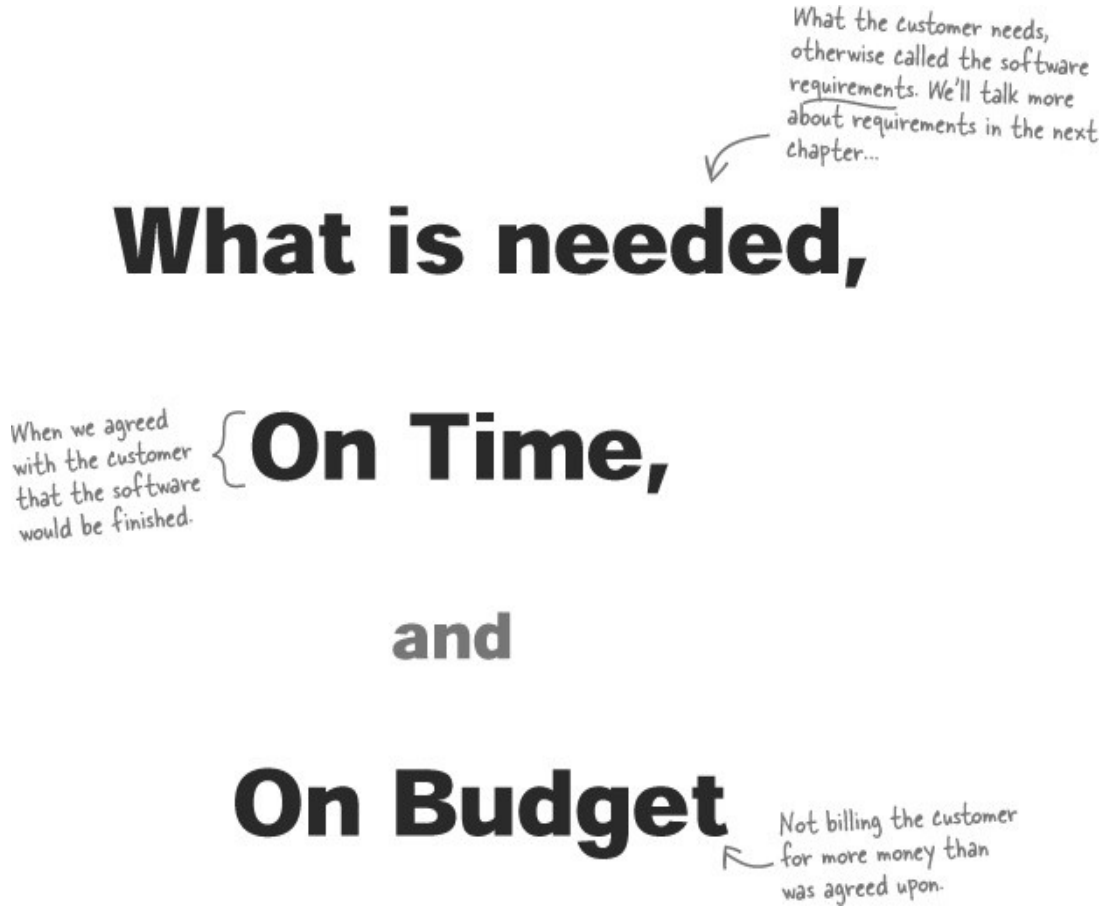
"Someone calling themselves a customer says they want something called service."

- Remember the #1 principle behind the Agile Manifesto?
  - "Our highest priority is to satisfy the customer through early and continuous delivery of valuable software." [MANIFESTO]

How To Make Your Customer Happy

- Correctly set customer expectations
  - How much will it cost?
    - No surprise here
    - Most customers want to figure out how much cash they'll have to spend
  - How long will it take?
    - The other big constraint is time.
    - Almost no customer ever says,

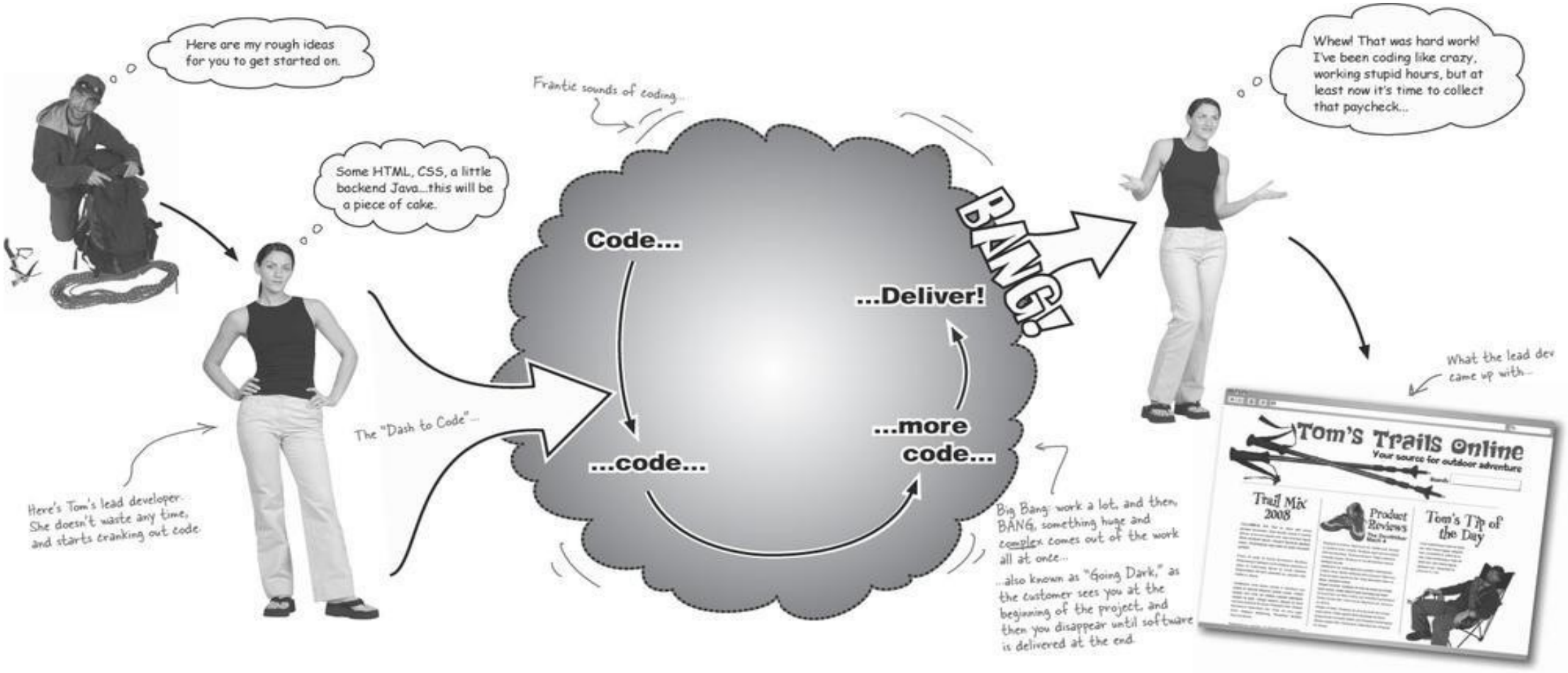
    "Take as long as you want!"
- Deliver what the customer needs



Big Bang

- Gather an idea of what the customer wants
- "Go dark"
  - Step 1: code
  - Step 2: code
  - ...
  - Step 52: code
  - Step 53: deliver

BIG BANG


Here are my rough ideas for you to get started on.
Some HTML, CSS, a little backend Java...this will be a piece of cake.
Frantic sounds of coding...
Code...
...code...
...Deliver!
...more code...
BANG!
The "Dash to Code"...
Here's Tom's lead developer. She doesn't waste any time, and starts cranking out code.
Big Bang work a lot, and then BANG, something huge and complex comes out of the work all at once...
...also known as "Going Dark," as the customer sees you at the beginning of the project, and then you disappear until software is delivered at the end.
Whew! That was hard work! I've been coding like crazy, working stupid hours, but at least now it's time to collect that paycheck...
What the lead dev came up with...
Tom's Trails Online
Your source for outdoor adventure
Trail Mix 2008
Product Reviews
Tom's Tip of the Day

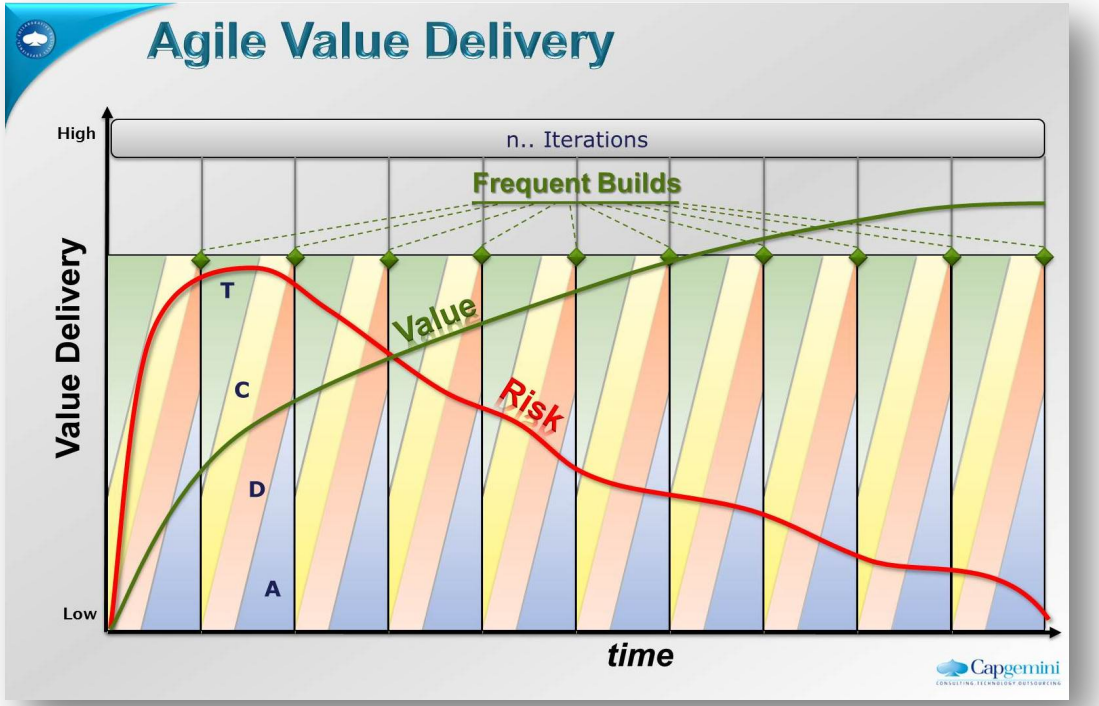
UNHAPPY CUSTOMER

- If your customer isn't happy, you built the wrong software
- Big bang software usually means working a whole lot,
- But it also means not showing the customer much until your work is done
  - The risk with that approach is you **think** you're building what the customer wants with no real feedback until you **think** you're finished
  - How do you figure out what the customer really wants?
    - It's not always easy…

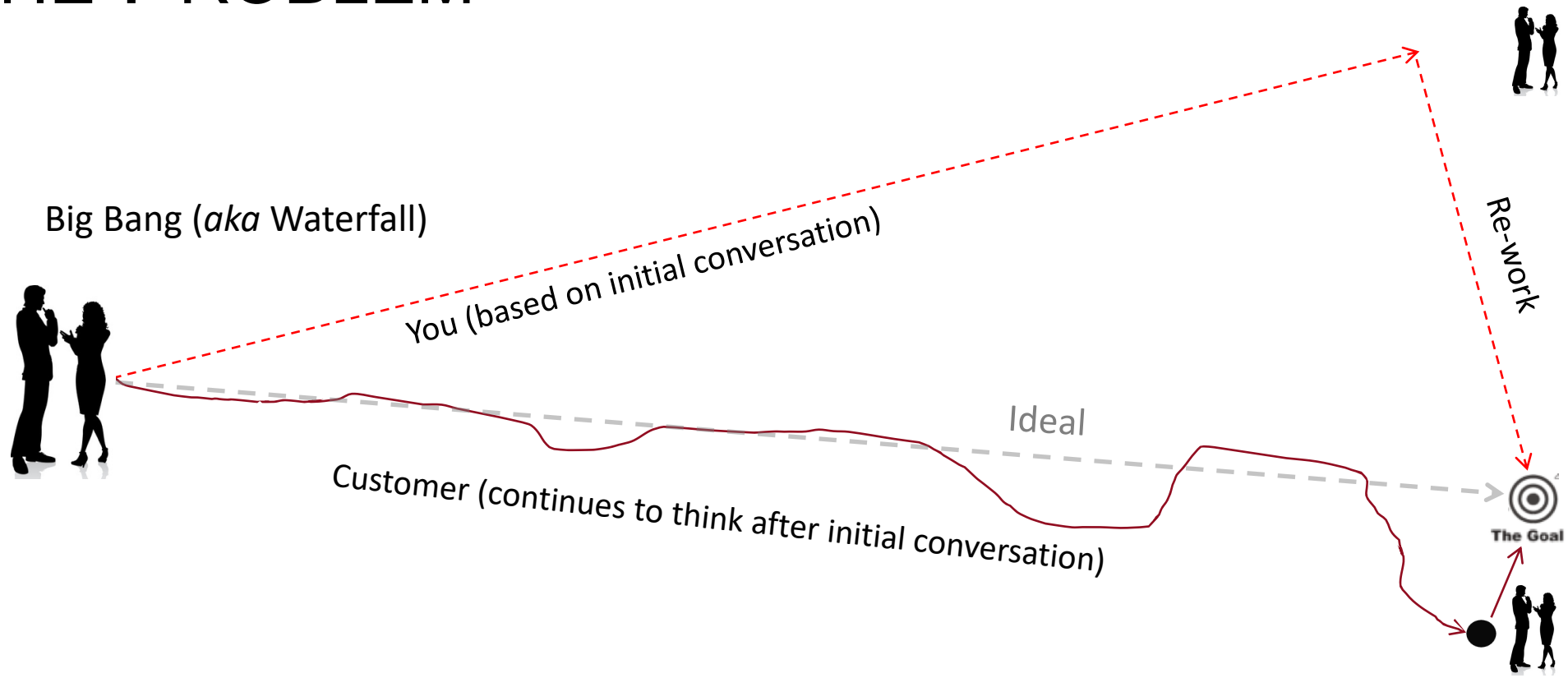
AGENDA

- Free Advice
- Review
- Happy Customer
- Iteration
- High-Level Requirements (features)
- Iteration Plan

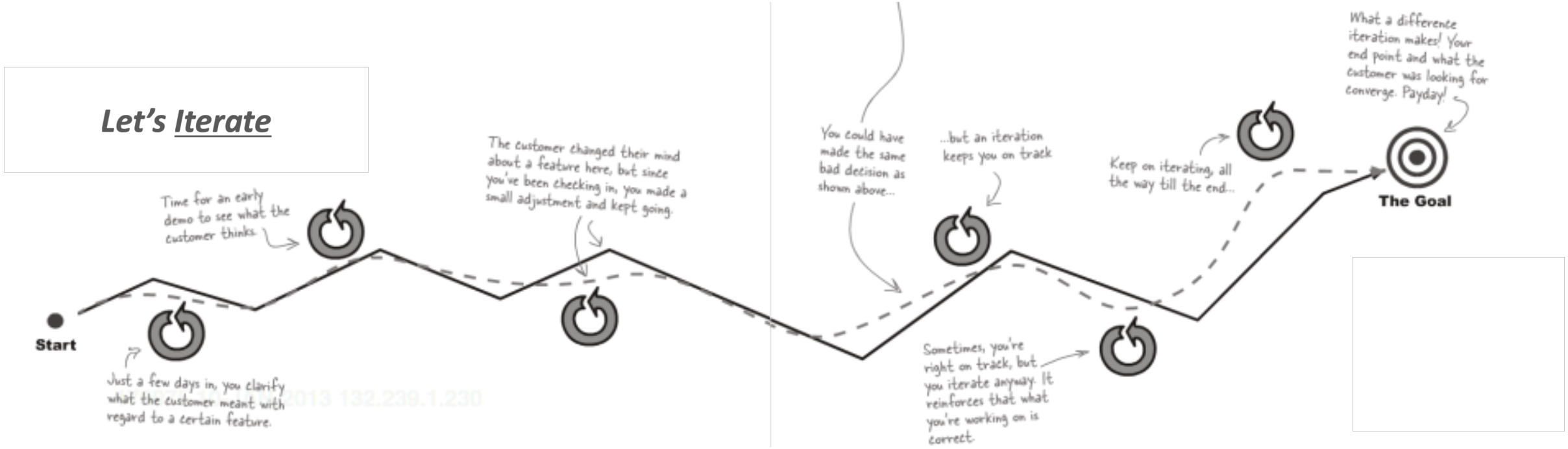
ITERATION - THE HEART OF AGILE PROCESS



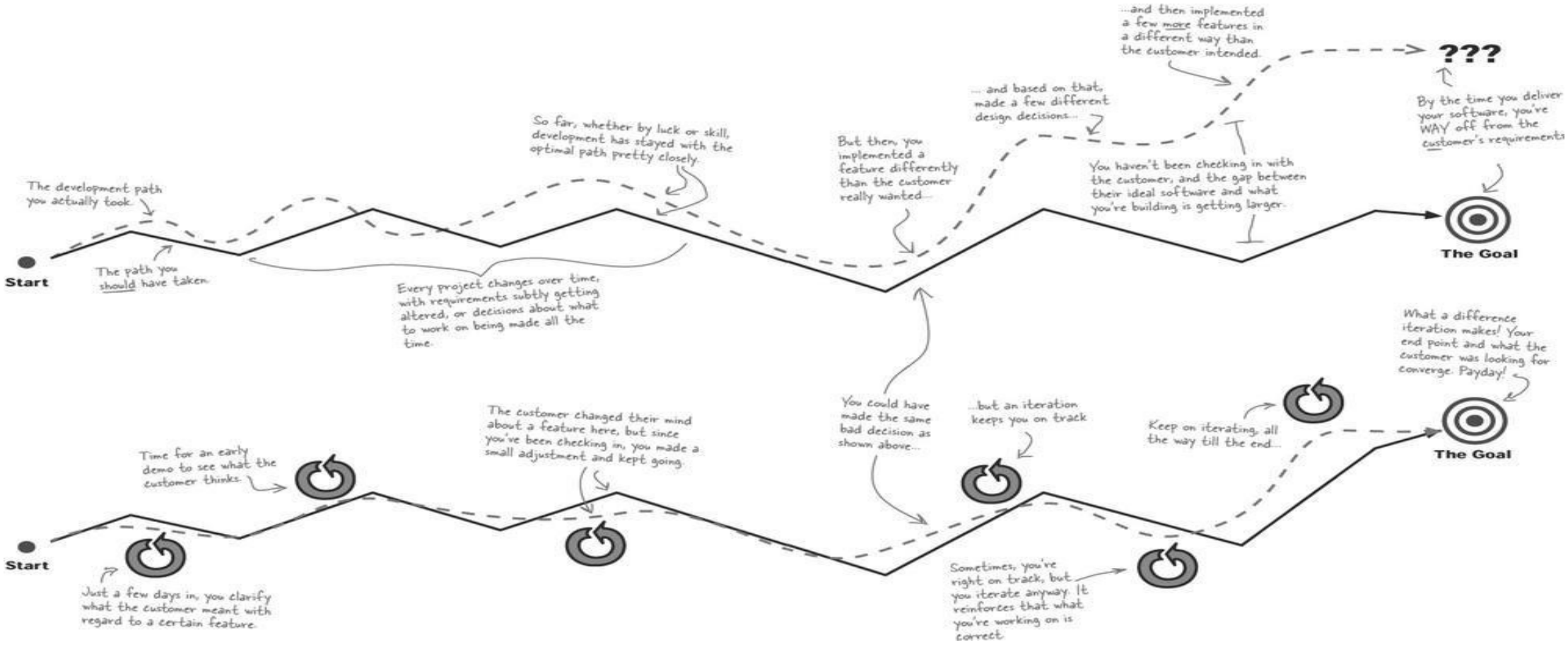
THE PROBLEM


Big Bang (*aka* Waterfall)
You (based on initial conversation)
Re-work
Ideal
Customer (continues to think after initial conversation)
The Goal

THE SOLUTION

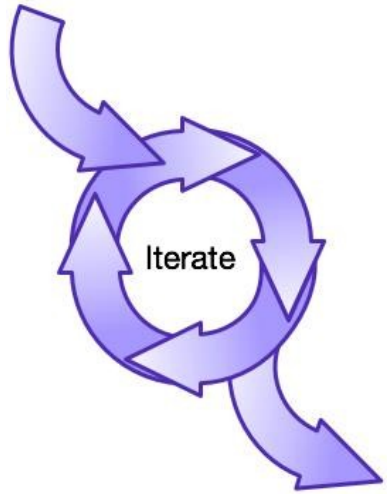


NO ITERATIONS VS. ITERATIONS



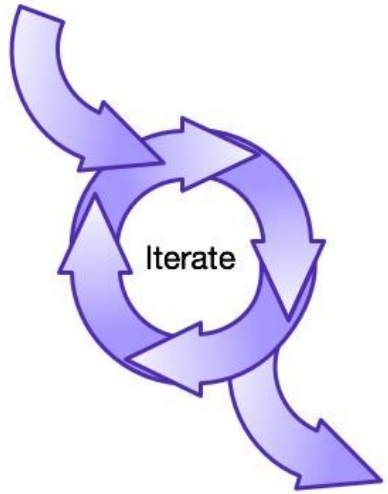
ITERATE INSTEAD

1. Gather **R**equirements
2. **D**esign Solution
3. Code Product (**C**onstruction)
4. Test Product (**T**esting)
5. Go To [1] Step #1



[1] Edsger Dijkstra would not like this step https://dl.acm.org/doi/10.1145/362929.362947

ITERATE INSTEAD



1. Gather **R**equirements
2. **D**esign Solution
3. Code Product (**C**onstruction)
4. Test Product (**T**esting)
5. Deliver Working Software
   - "Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale." [MANIFESTO]
6. Go To Step #1

- **Each iteration is a mini-project**

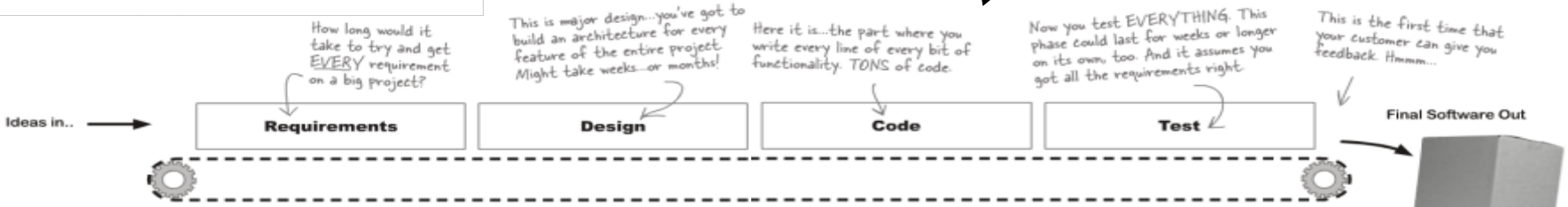
ITERATION

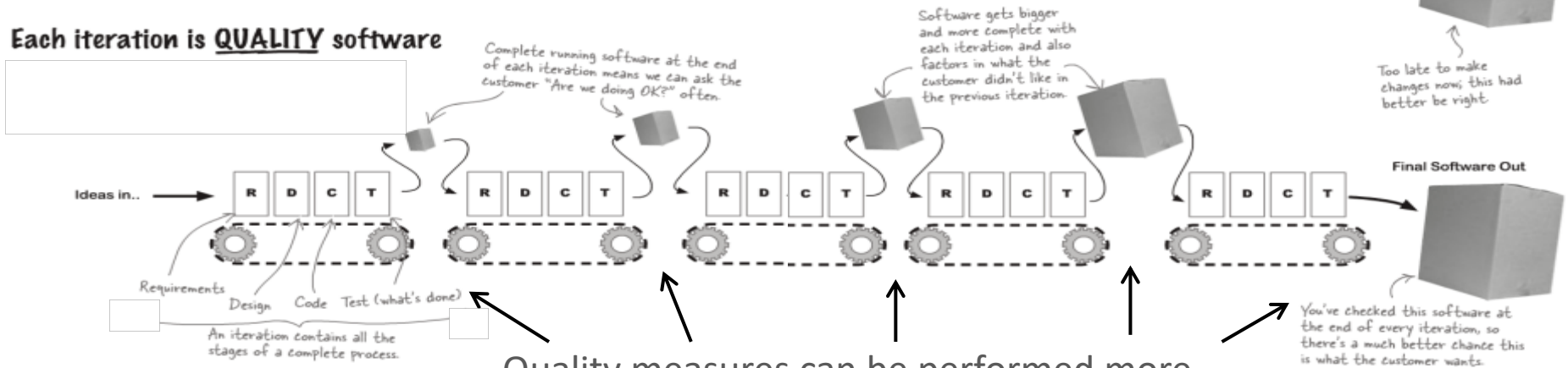
- Is iteration only helpful when I am
  - working in a team?
  - working on a big project?
  - working on a long term project?
- Each iteration is a mini-project
- How do we make an iteration plan?

ITERATIONS ARE MINI-PROJECTS

"Big Bang" Waterfall Development

Few **well-defined** quality measurement points, little data to guide process improvement





Quality measures can be performed more frequently, informing frequent adjustments

AGENDA

- Free Advice
- Review
- Happy Customer
- Iteration
- High-Level Requirements (features)
- Iteration Plan

Iteration Planning

- Gather high-level **requirements** ("features" in chapter 1)
  - Priority
    - Customer prioritizes each feature
  - Time to complete
    - You estimate how long each feature will take to code

Iteration Period

- Pick a length of time for each iteration
- Book suggests around 20 working days
  - (one calendar month)

DEPENDENCIES

- What about feature dependencies?
- You must plan for this; place dependent tasks in the same iteration

AGENDA

- Free Advice
- Review
- Happy Customer
- Iteration
- High-Level Requirements (features)
- Iteration Plan

Iteration Planning

- Now we can build a plan based on time and priority
  - Higher priority features are worked on first
  - How many features can we fit into an iteration?
    - Add some shorter-duration features to fill time

ESTIMATE & PRIORITIZE

- **You** estimate how long each feature will take to code
- Customer prioritizes each feature

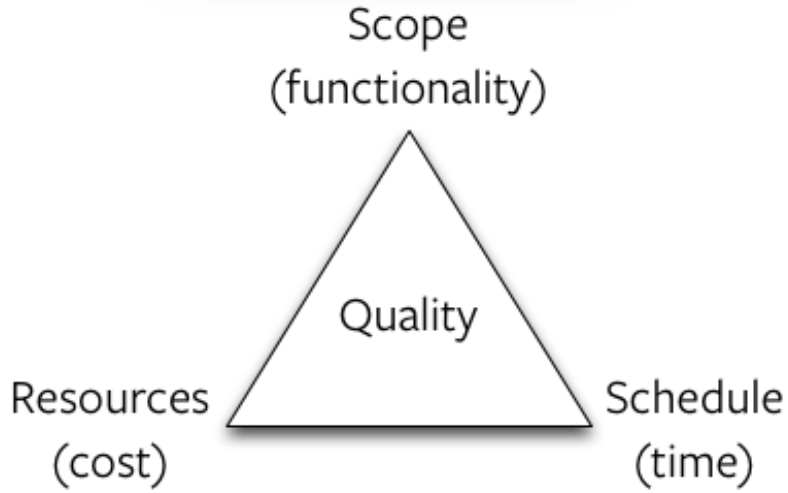
ESTIMATE & PRIORITIZE

- You estimate how long each feature will take to code
- **Customer** prioritizes each feature

Iteration Planning - With the Customer

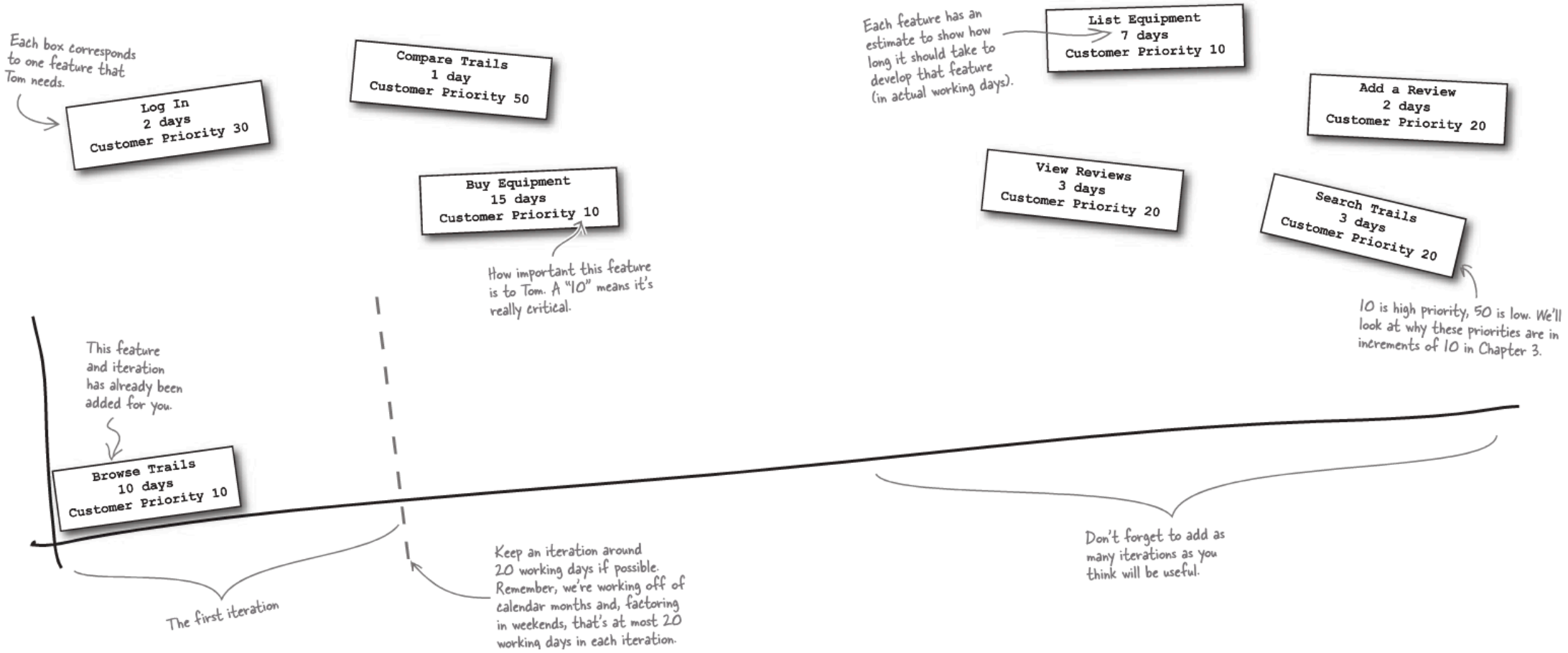
- With customer, you work out:
  1. List of requirements or features
  2. Budget
  3. Deadline

- This triangle has a couple of names:
  1. Project management triangle
  2. Iron triangle

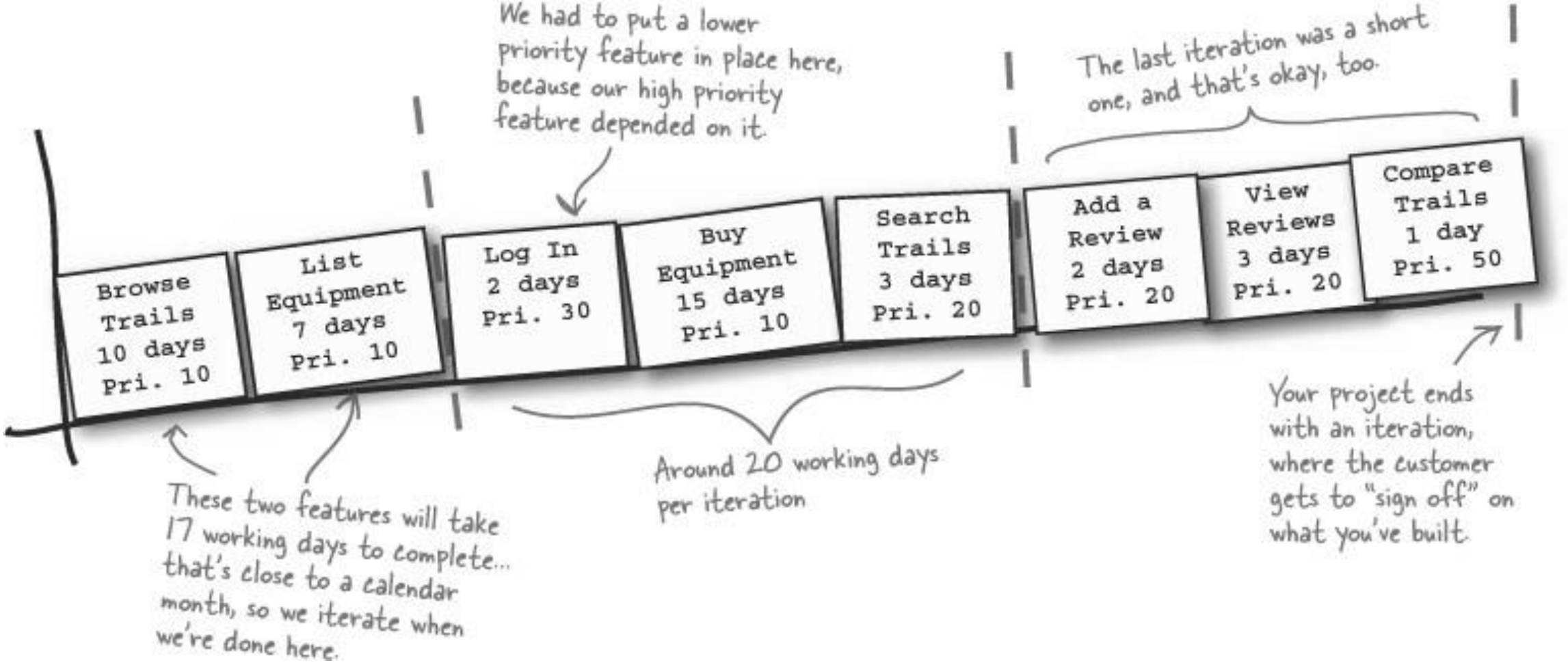


Exercise Intro

ESTIMATE & PRIORITIZE

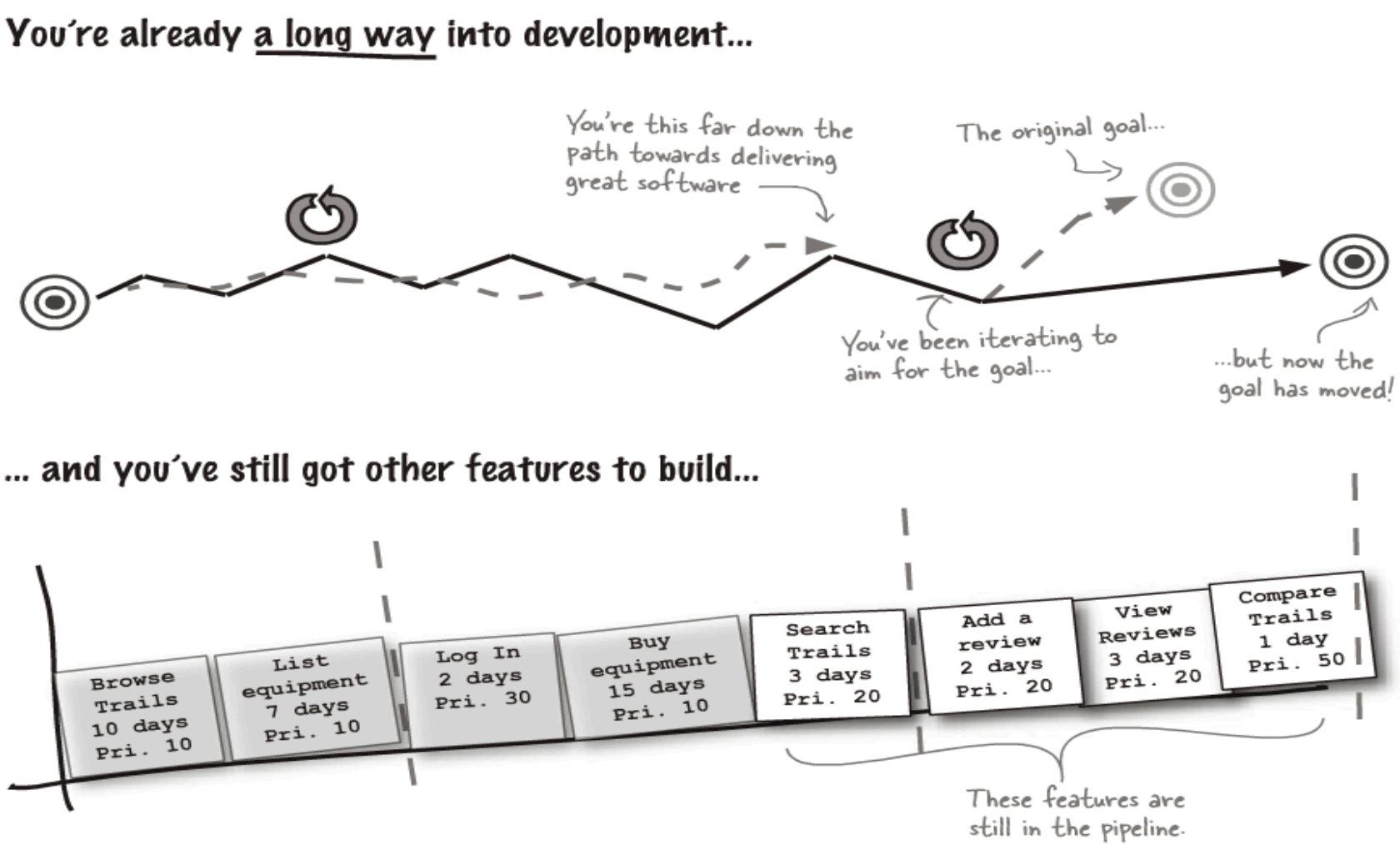


POTENTIAL SOLUTION



EXPECT CHANGE!

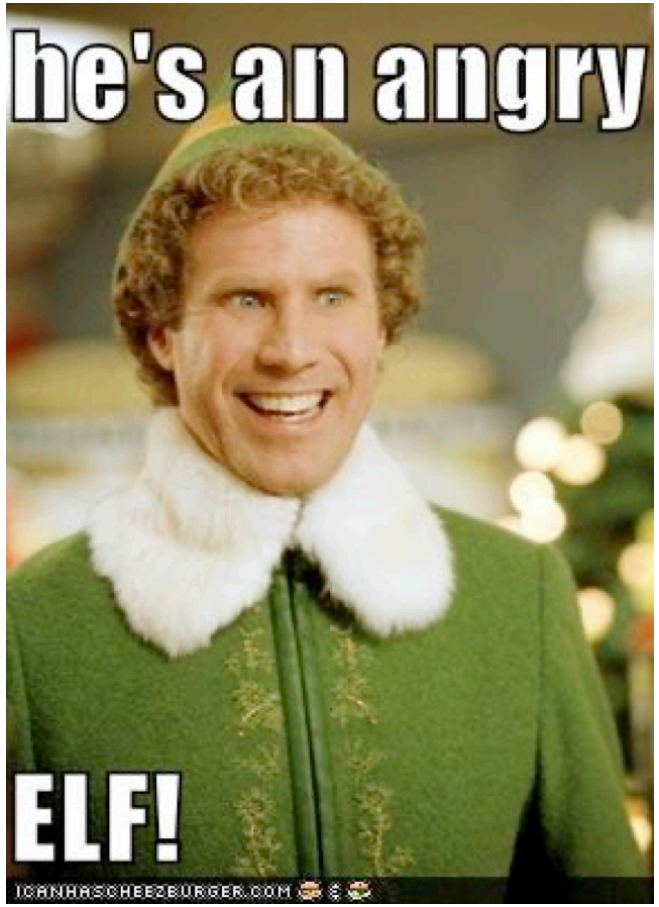
NEED TO ADJUST!



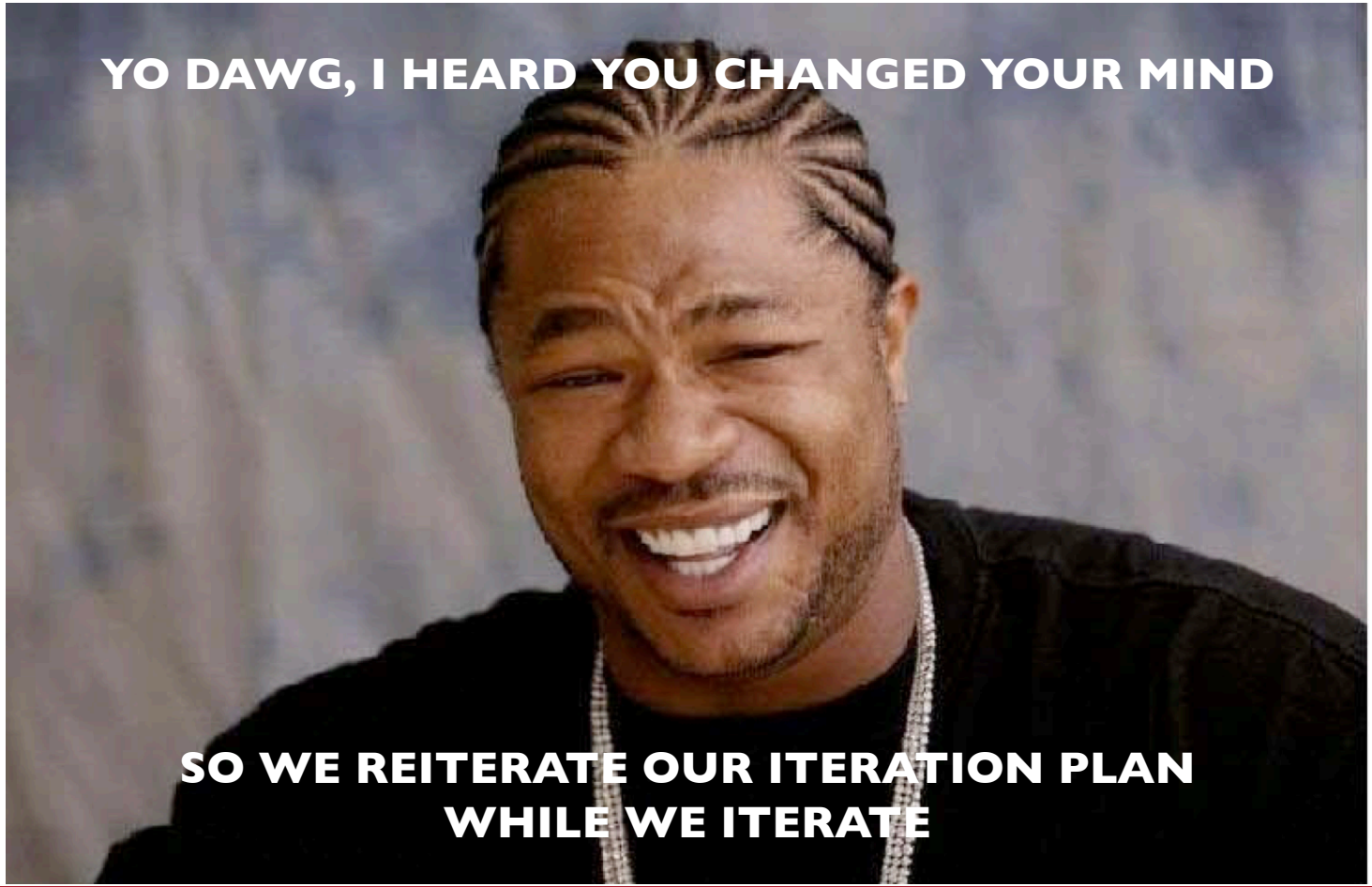
NEED TO ADJUST … BUT DEADLINE IS THE SAME ?

When Features Change

- They will!
- When features change (angry elves call this "scope creep")
  - Adjust your budget!
    - time and
    - cost

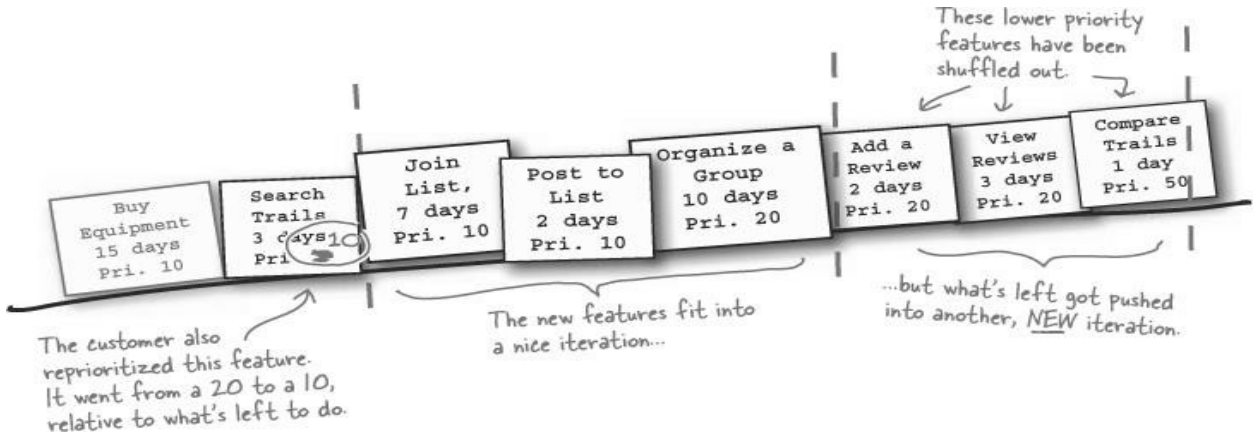


REITERATE YOU ITERATION PLAN



REITERATE YOU ITERATION PLAN

1. Estimate the new features
2. Have your customer prioritize the new features
3. Rework your iteration plan
4. Check your project deadline



1 Estimate the new features

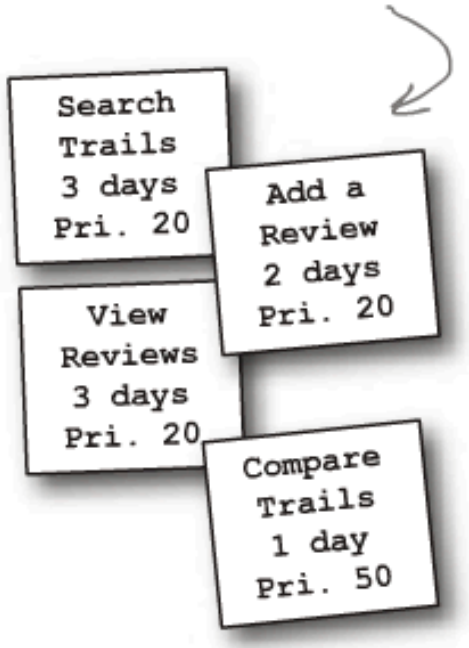
First, you need to estimate how long each of the new features is going to take. We'll talk a lot more about estimation in a few chapters, but for now, let's say we came up with these estimates for the three new features:

2 Have your customer prioritize the new features

Tom already gave everything a priority of "20," right? But you really need him to look at the other features left to implement as well, and prioritize in relation to those.

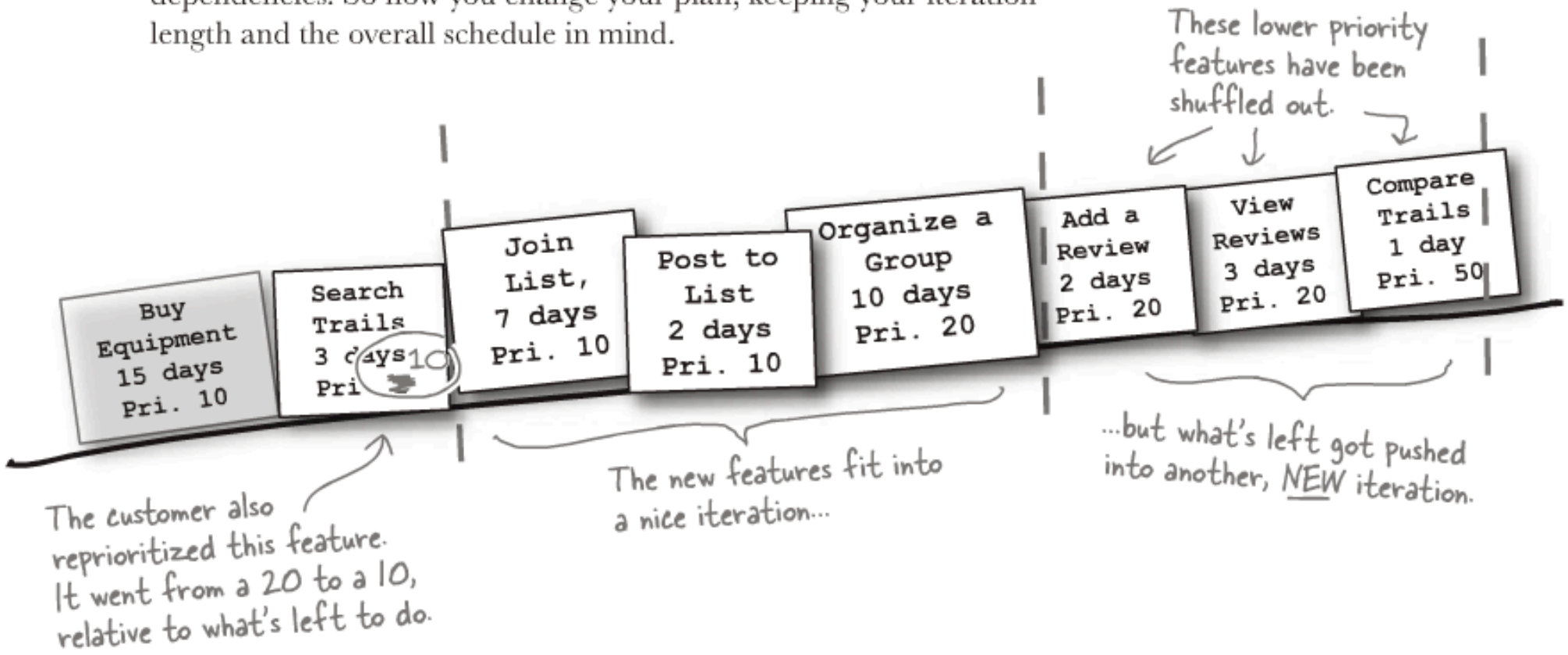
Tom decided that two of these are more important than anything that's left, so they get a 10. The other feature is a 20, and can be mixed in.

A priority of 20, relative to these remaining features means we can sprinkle one more feature in anywhere before comparing trails.



3 Rework your iteration plan

The ordering is set based on prioritization, and there aren't any dependencies. So now you change your plan, keeping your iteration length and the overall schedule in mind.



4 Check your project deadline

Remember the TrailMix Conference? You need to see if the work you've got left, including the new features, still can get done in time. Otherwise, Tom's got to make some hard choices.

Work left to do

(Days of work left)

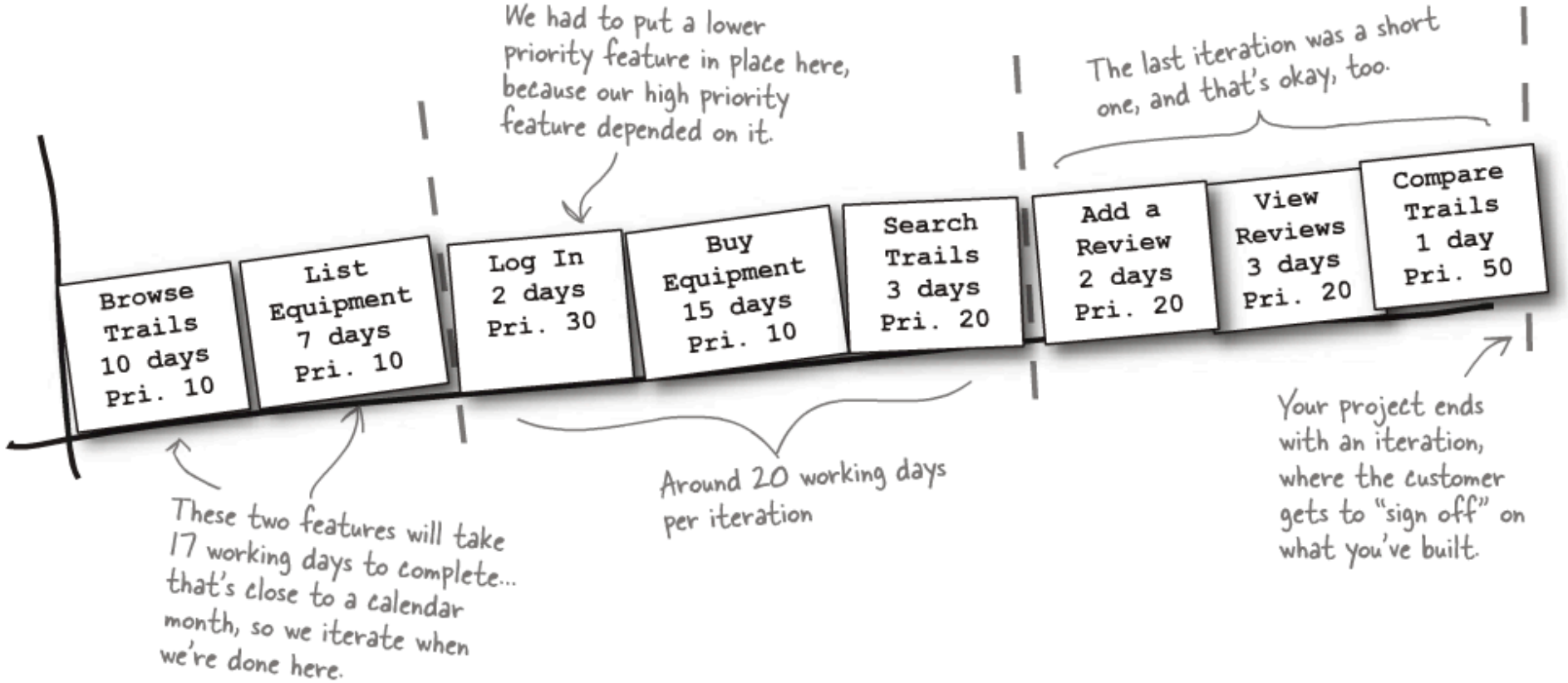
−

(Days left before deadline)

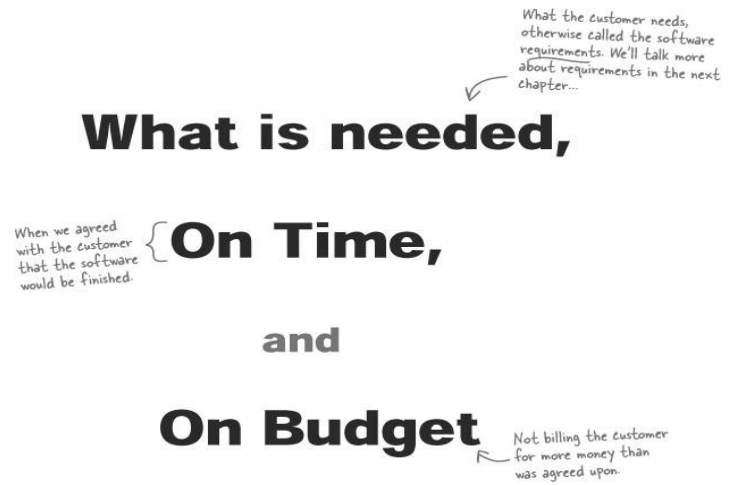
Days before TrailMix Con

= Can you do it?

If this number is negative, you're in good shape.


We had to put a lower priority feature in place here, because our high priority feature depended on it.
The last iteration was a short one, and that's okay, too.
Browse Trails 10 days Pri. 10
List Equipment 7 days Pri. 10
Log In 2 days Pri. 30
Buy Equipment 15 days Pri. 10
Search Trails 3 days Pri. 20
Add a Review 2 days Pri. 20
View Reviews 3 days Pri. 20
Compare Trails 1 day Pri. 50
These two features will take 17 working days to complete... that's close to a calendar month, so we iterate when we're done here.
Around 20 working days per iteration
Your project ends with an iteration, where the customer gets to "sign off" on what you've built.

RECAP



Any concerns about this?

RECAP

- What do you need for planning?
  - High-level requirements
    - Prioritized
    - Time to develop each