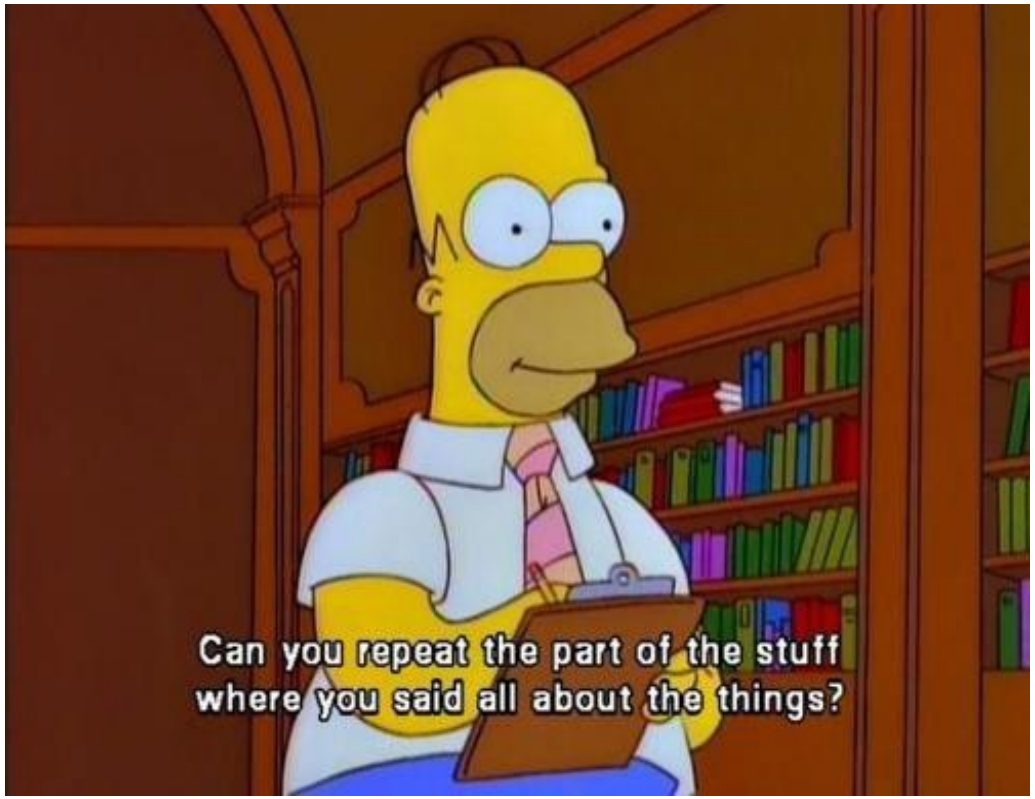
USER STORIES & TASKS

Content from Chapter 4 of "Head First Software Development", Pilone et al.

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

Questions From Last Class?

- Planning
- Prioritization
- Velocity
  - Initial value?
- Burndown Chart
  - X vs. Y axis
  - Units



AGENDA

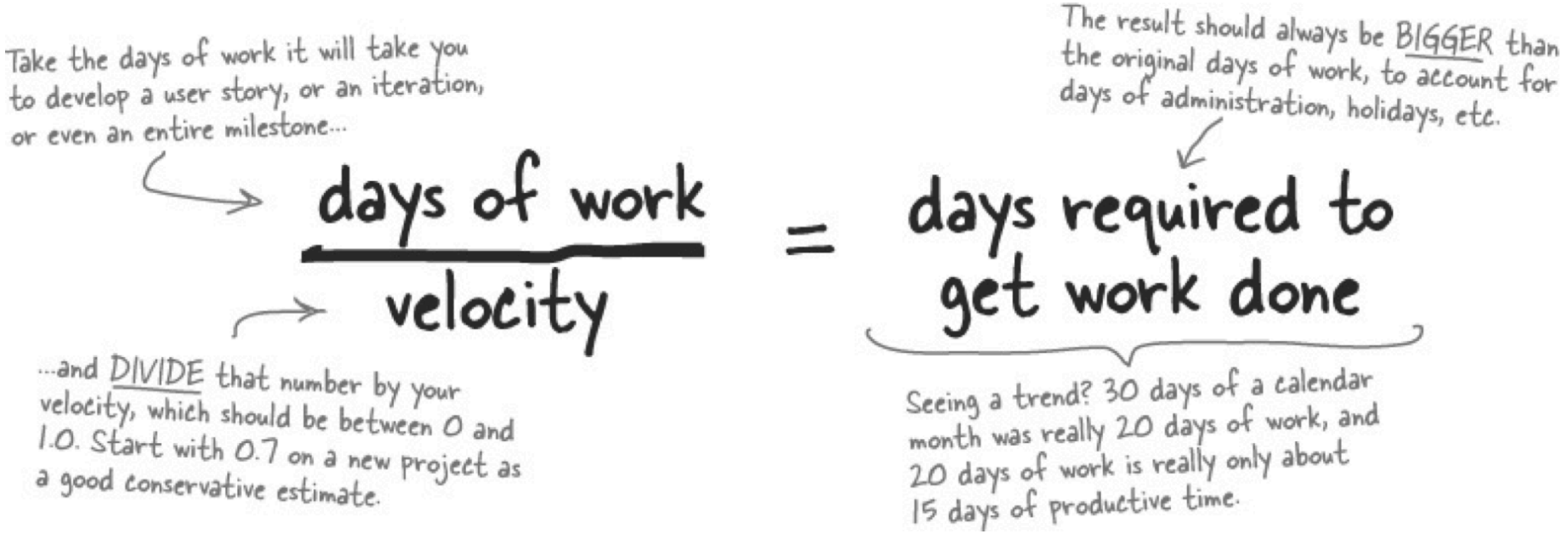
- Review
  - Velocity
  - Burn-Down Chart
- User Stories
  - What to do with them
- Tasks
  - Estimating with tasks
  - Multitasking

AGENDA

- Review
  - Velocity
  - Burn-Down Chart
- User Stories
  - What to do with them
- Tasks
  - Estimating with tasks
  - Multitasking

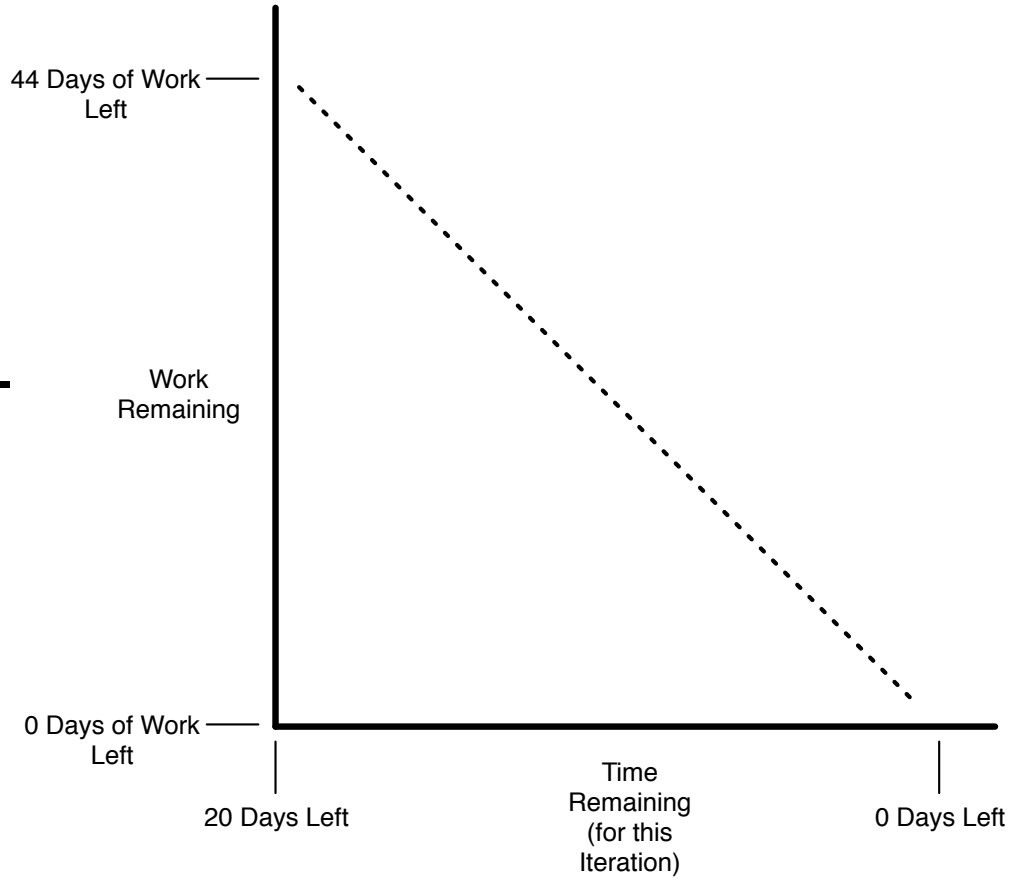
REVIEW - VELOCITY

- Helps predict the actual time it will take to complete work
- Velocity is a number between 0 and 1.0



REVIEW - BURN-DOWN

- Chart used to visually show progress and may help to reprioritize tasks
- The dotted line shown on the *Burn-Down Chart* represents the ideal burn-down rate.
- What it means for an iteration if you are plotting your burn-down *below* the dotted line?

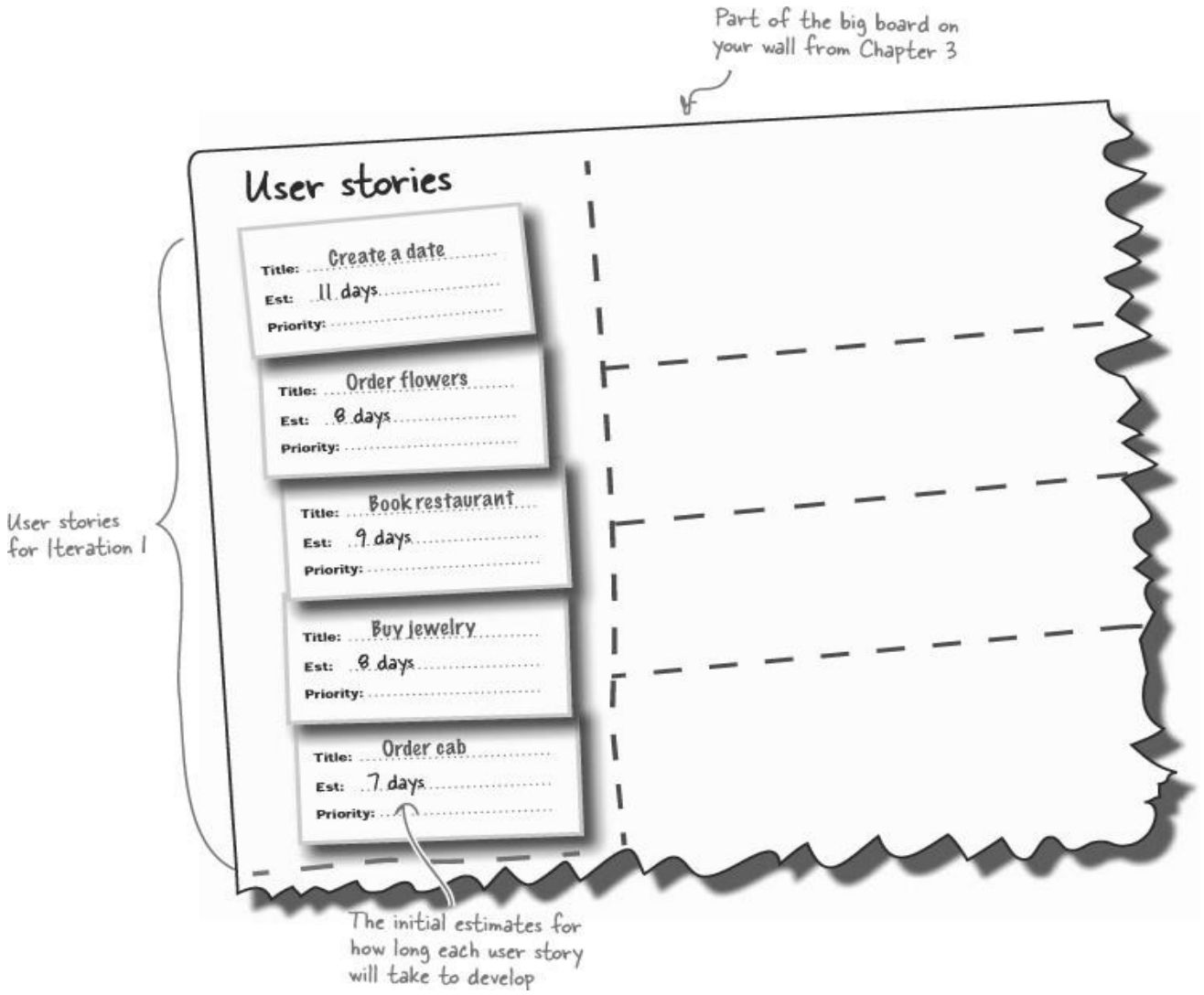


AGENDA

- Review
  - Velocity
  - Burn-Down Chart
- User Stories
  - What to do with them
- Tasks
  - Estimating with tasks
  - Multitasking

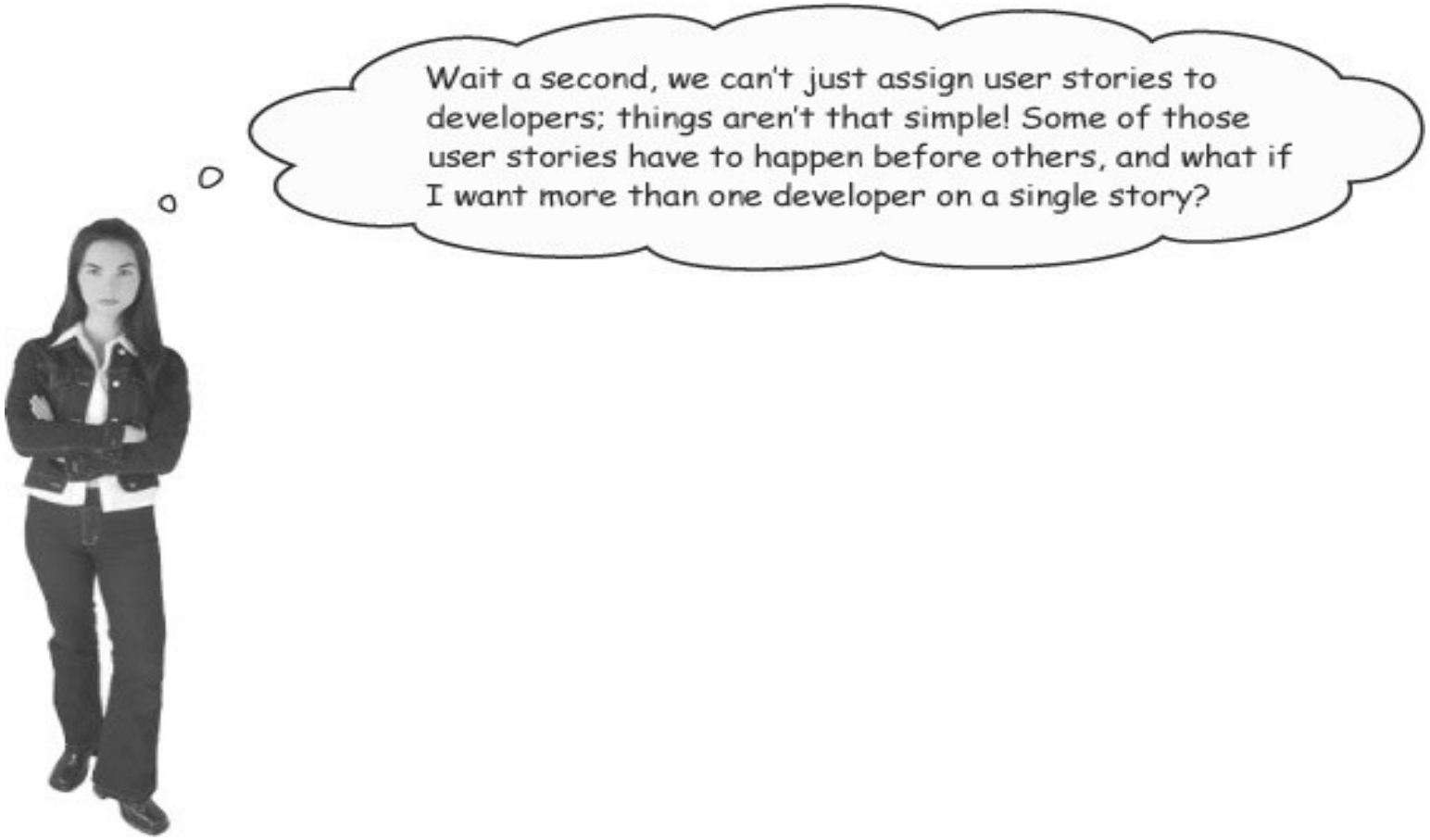
THE BIG BOARD

- We Have Our Stories, Now What?



JUST ASSIGN DEVELOPERS FOR EACH STORY?

- No!
- Why?



Granularity Of Work Vs. Stories

- User stories are for users.
  - Described what the software needs to do.

- Coding may need to break down stories
  - Each story consists of Tasks.
  - Task = small bit of functionality
  - Set of tasks = 1 user story.

AGENDA

- Review
  - Velocity
  - Burn-Down Chart
- User Stories
  - What to do with them
- Tasks
  - Estimating with tasks
  - Multitasking

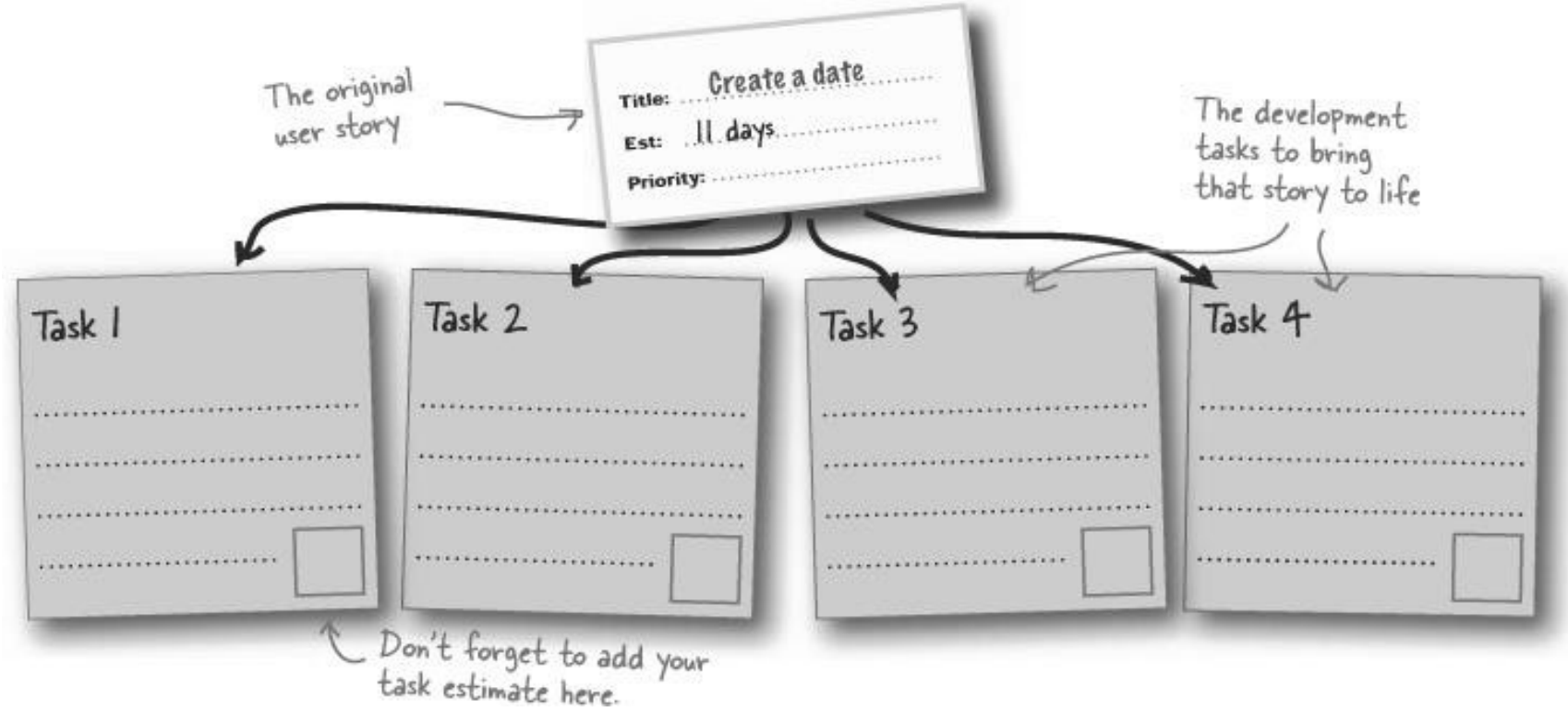
TASKS

- Specific pieces of development work for 1 developer (or pair programming group)

- Tasks each have
  - Title
  - Rough description (of how task will be completed)
  - Estimate -> Planning Poker!

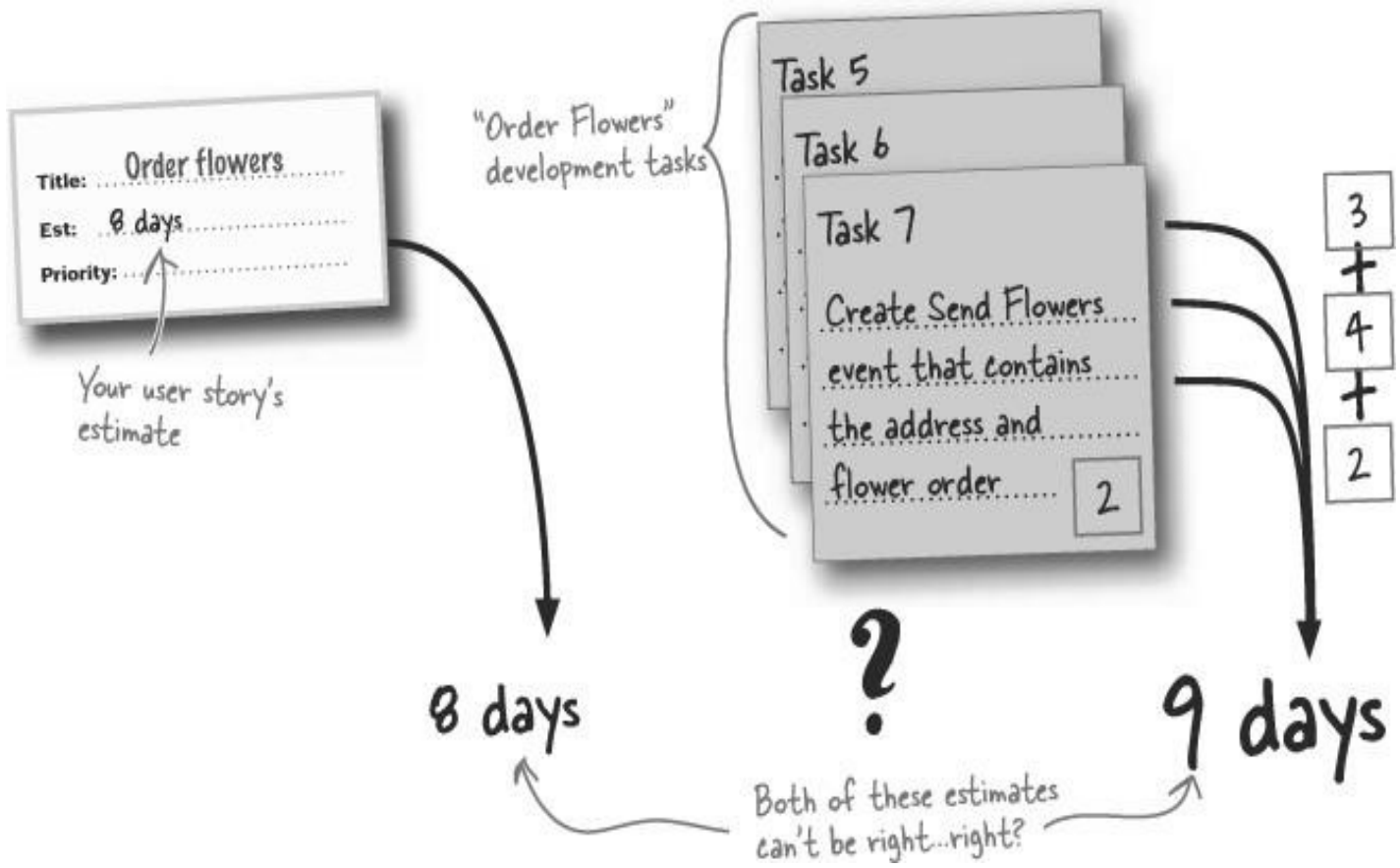
EXAMPLE EXERCISE

- User Story: Creating a date (in a desktop day/date planner)



BEFORE THAT THOUGH

- We are now breaking down our estimates
- What if they do not correspond to total estimate of story?

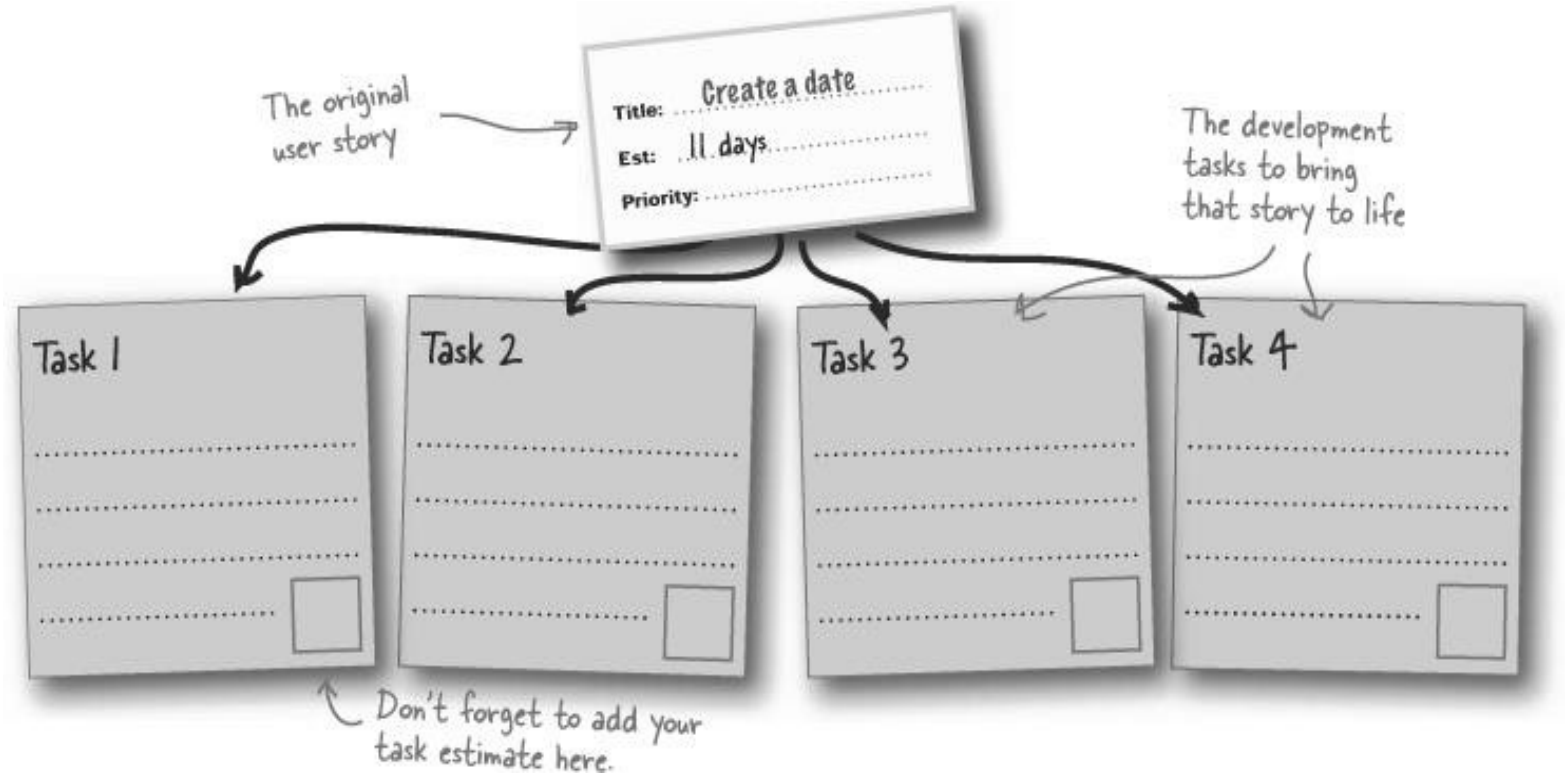


Task Estimates Add Confidence

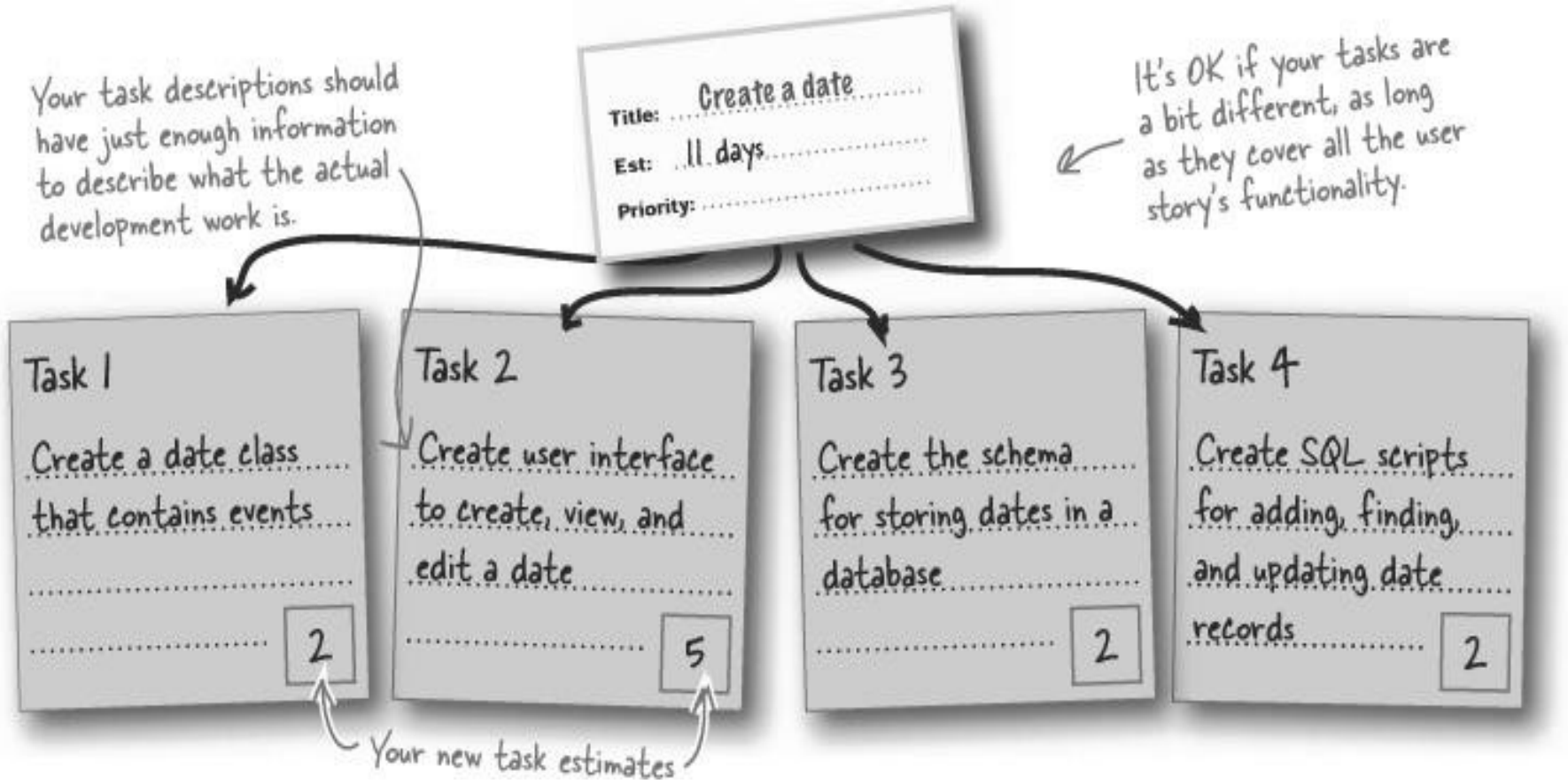
- Tasks are another level of detail (less abstract)
- Always best to rely on task estimates
  - Give that to your customer originally
- Add confidence to the story estimates. Earlier >>>

EXAMPLE EXERCISE WITH PLANNING POKER

- User Story: Creating a date (in a desktop day/date planner)

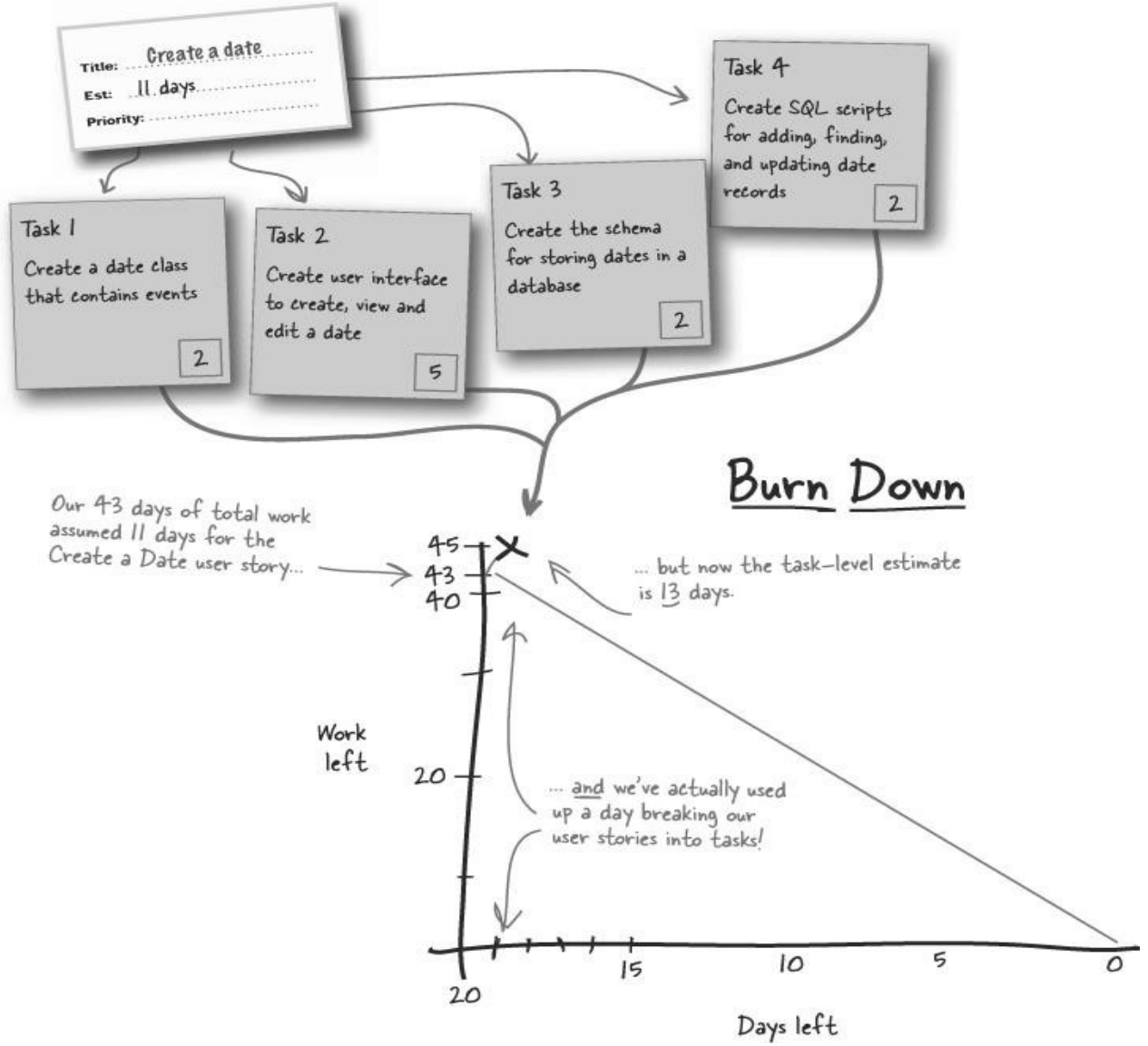


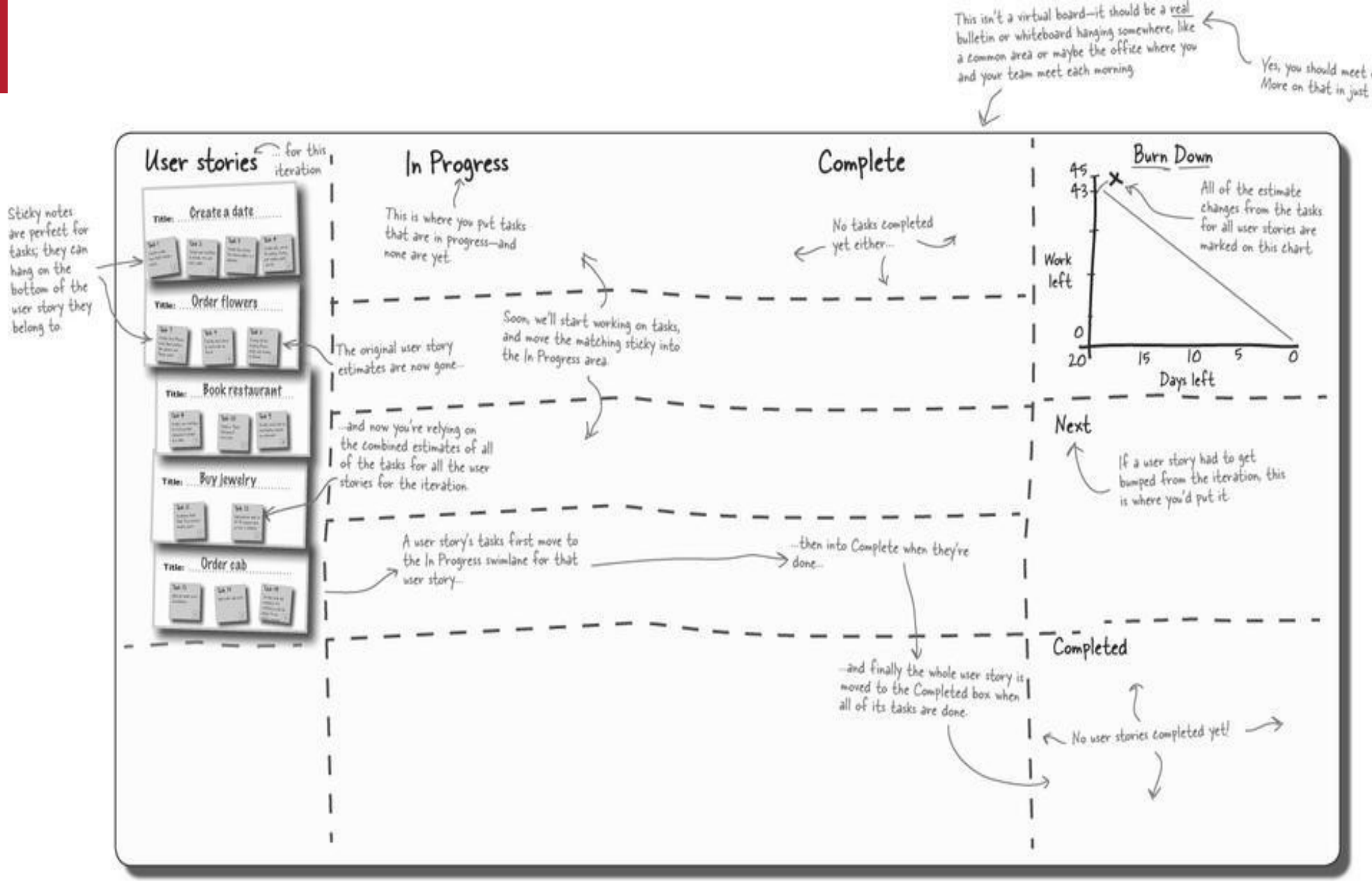
SAMPLE SOLUTION



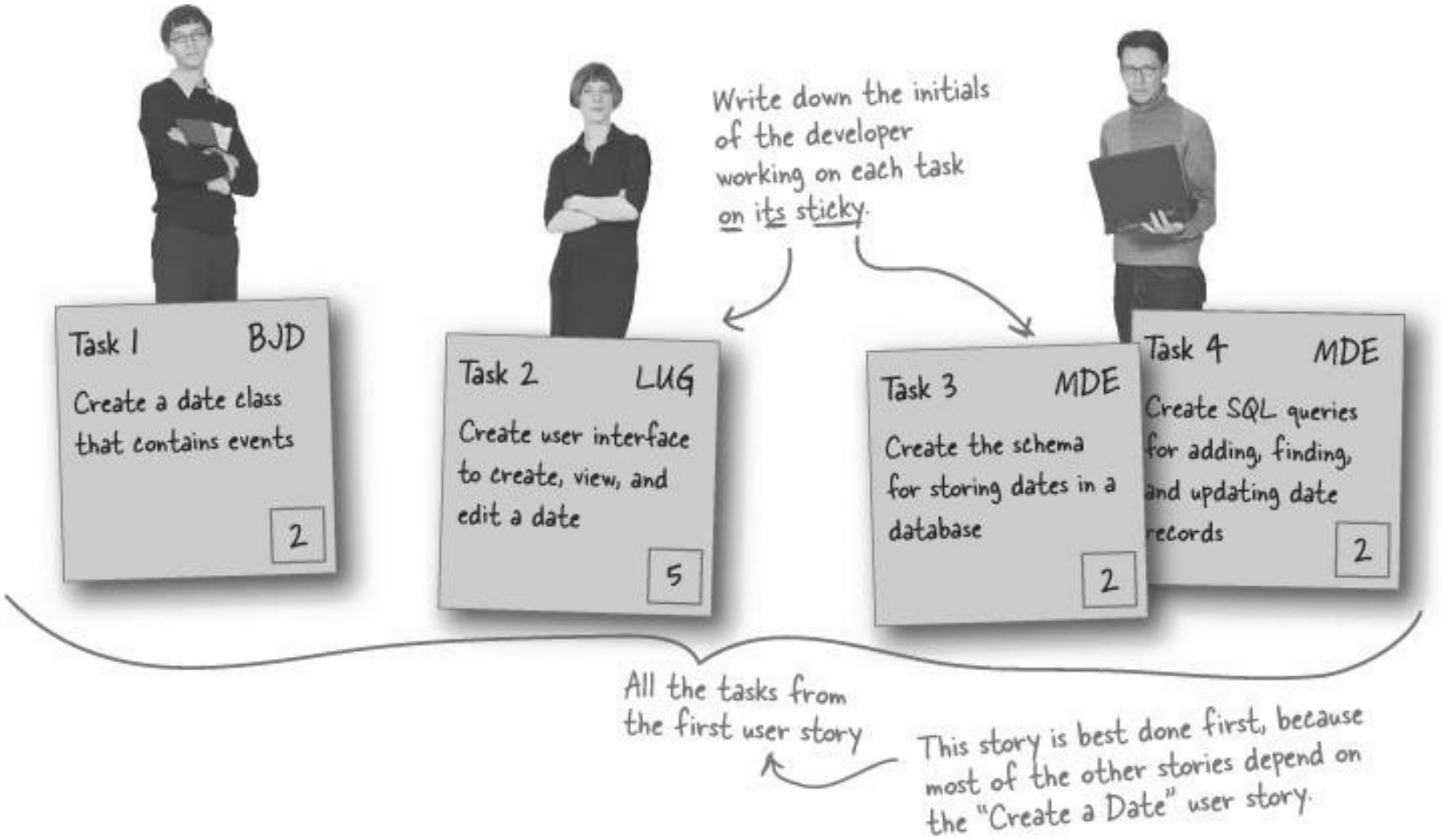
Burn Down Is Back

- Track **only** the work left
  - Every time there's work or estimate review.


Title: Create a date
Est: 11 days
Priority:
Task 1
Create a date class that contains events
2
Task 2
Create user interface to create, view and edit a date
5
Task 3
Create the schema for storing dates in a database
2
Task 4
Create SQL scripts for adding, finding, and updating date records
2
Burn Down
Our 43 days of total work assumed 11 days for the Create a Date user story...
... but now the task-level estimate is 13 days.
... and we've actually used up a day breaking our user stories into tasks!
Work left
45
43
40
20
20
15
10
5
0
Days left

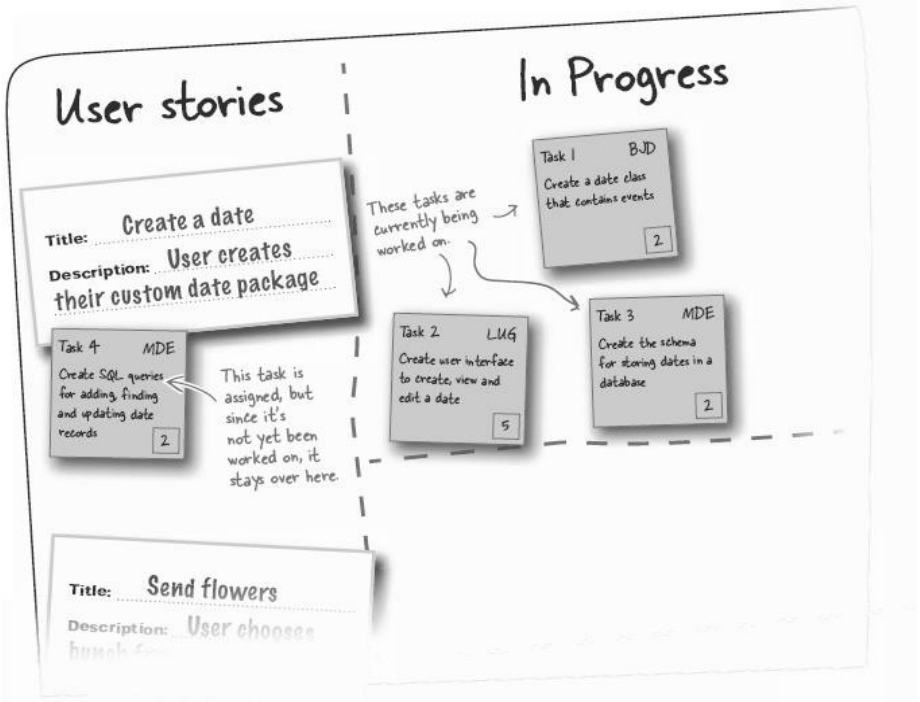

This isn't a virtual board—it should be a real bulletin or whiteboard hanging somewhere, like a common area or maybe the office where you and your team meet each morning.
Yes, you should meet each morning! More on that in just a minute...
User stories
for this iteration
In Progress
Complete
Burn Down
Sticky notes are perfect for tasks; they can hang on the bottom of the user story they belong to.
Title: Create a date
Title: Order flowers
Title: Book restaurant
Title: Buy jewelry
Title: Order cab
This is where you put tasks that are in progress—and none are yet.
No tasks completed yet either...
45
43
Work left
0
20
15
10
5
0
Days left
All of the estimate changes from the tasks for all user stories are marked on this chart.
The original user story estimates are now gone...
Soon, we'll start working on tasks, and move the matching sticky into the In Progress area.
...and now you're relying on the combined estimates of all of the tasks for all the user stories for the iteration.
Next
If a user story had to get bumped from the iteration, this is where you'd put it.
A user story's tasks first move to the In Progress swimlane for that user story...
...then into Complete when they're done...
...and finally the whole user story is moved to the Completed box when all of its tasks are done.
Completed
No user stories completed yet!

Got To Work!


Write down the initials of the developer working on each task on its sticky.
Task 1 BJD
Create a date class that contains events
2
Task 2 LUG
Create user interface to create, view, and edit a date
5
Task 3 MDE
Create the schema for storing dates in a database
2
Task 4 MDE
Create SQL queries for adding, finding, and updating date records
2
All the tasks from the first user story
This story is best done first, because most of the other stories depend on the "Create a Date" user story.

In Progress

- In progress must only be designated when work is actually started.
  - Board is only as useful as it is precise.
- Move around sticky notes accordingly.
- Board ~= reality



MULTITASKING

- Sometimes tasks are related
  - Overlap => more work to complete independently.
- SOMEtimes working on both is the best option.
  - Work completed in one task can inform decisions for another task.
  - More efficient to work on both tasks at the same time.
- Try to double up tasks that are related
- Don’t double up on tasks that have large estimates though

STANDUP MEETINGS – STANDING >> SITTING

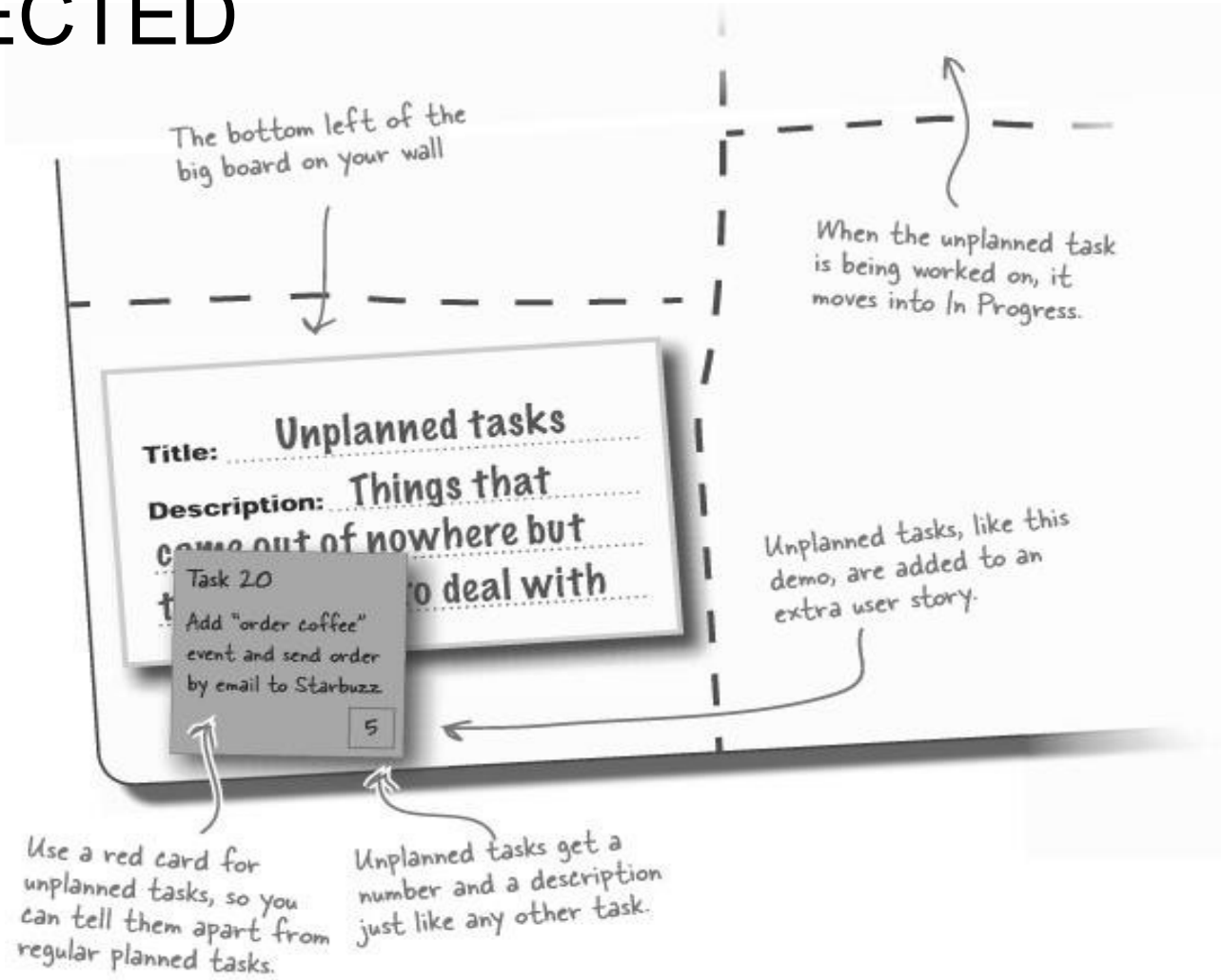
- A meeting so quick don't even need to sit
- Includes
  - Tracking progress
  - Update burn-down rate
  - Update tasks (completed or not)
  - Talk about yesterday and today
  - Bring up issues
- Last between 5 and 15 minutes
  - How not to… : https://www.youtube.com/watch?v=oLmDe8pAc6I

PLAN FOR THE UNEXPECTED

"Before you begin, I'd like to thank you for coming in early to do this on such short notice."

PLAN FOR THE UNEXPECTED

- Bottom left of the big board allocated for “Unplanned tasks”

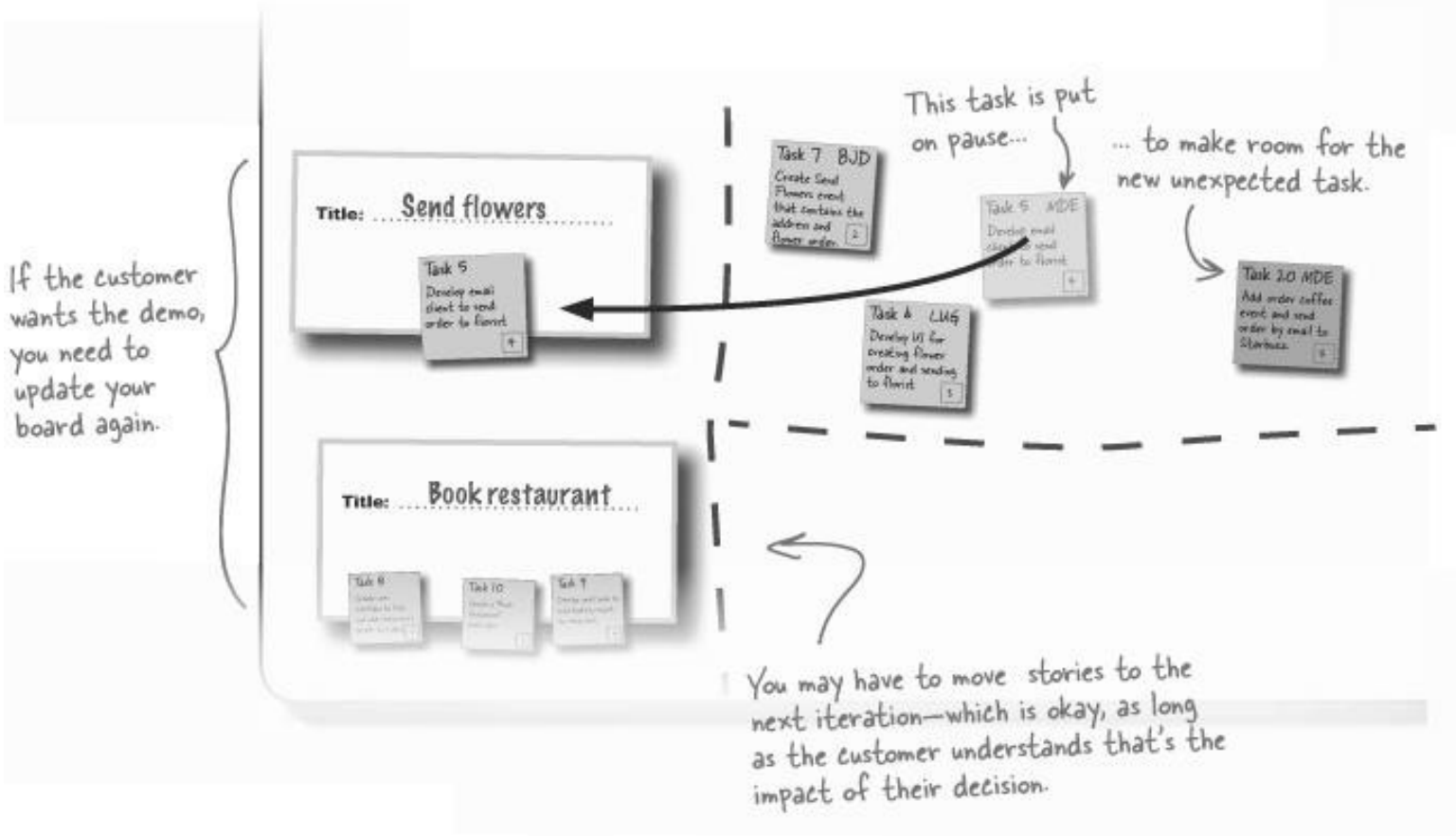


PLAN FOR THE UNEXPECTED – TALK TO THE CUSTOMER

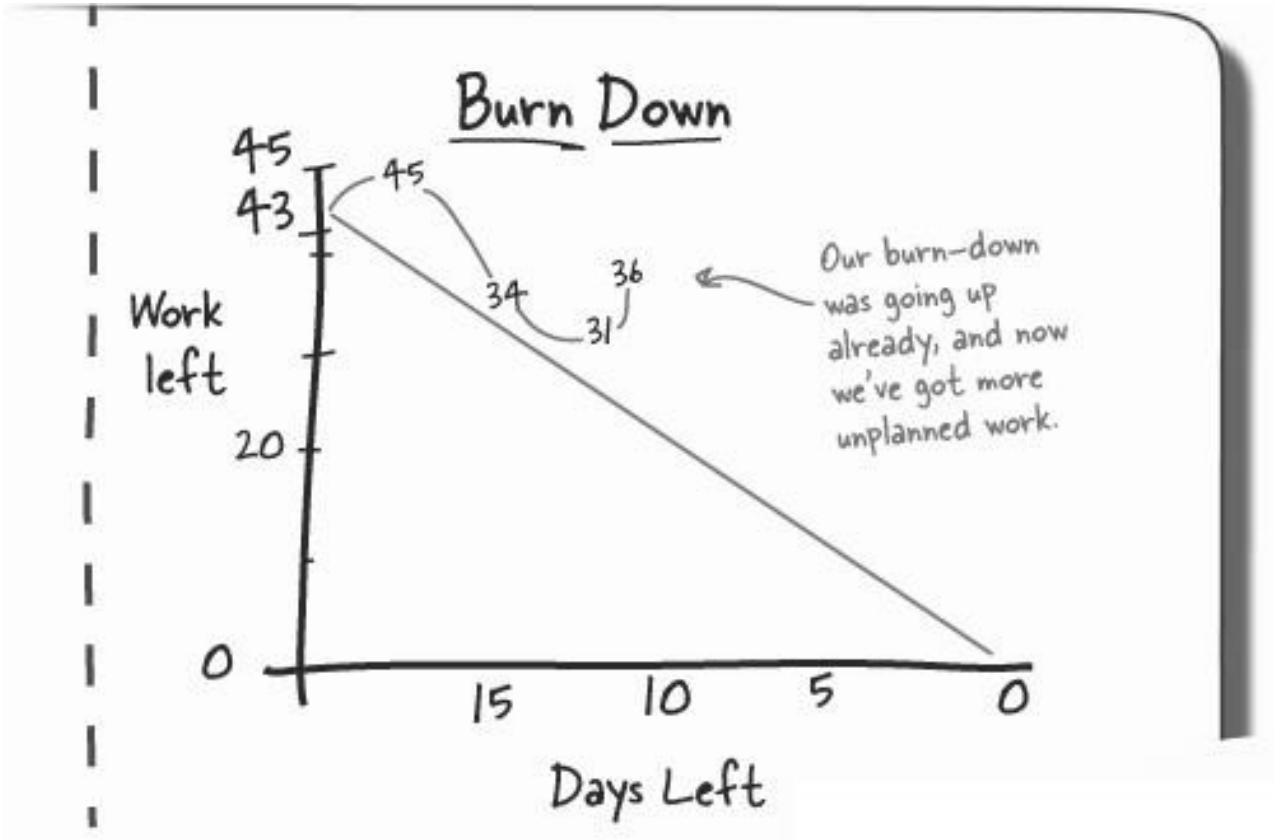
- Let the customer reset the priority
  - Estimate the amount of work the new task will take
  - Explain how it will affect the schedule.

- Give them all the information
  - Let them decide.
  - Not the end of the world. Just need to make sure customer understands impact

PLAN FOR THE UNEXPECTED – TALK TO THE CUSTOMER

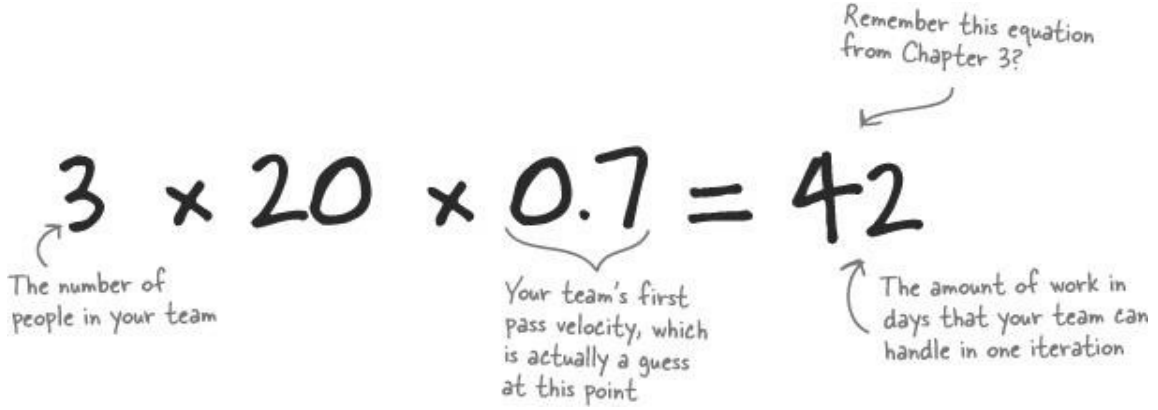


BURNDOWN RATE IMPACT

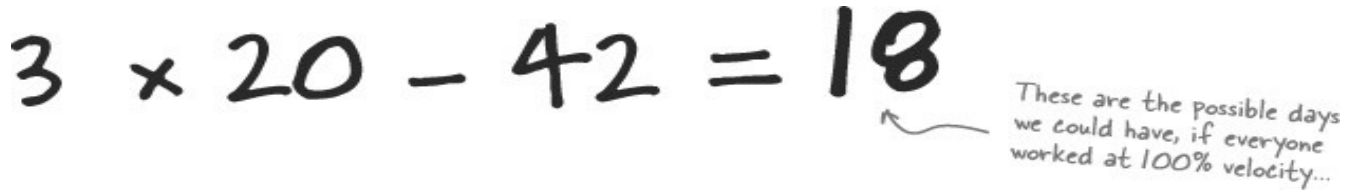


WHAT ABOUT VELOCITY + UNPLANNED?

- Well, we originally had



- So why not…



What About Velocity + Unplanned?

- Not applicable here.
- Velocity gauges team performance, not handling unplanned tasks.
- That “float” disappears quickly
  - Accidents/delays
  - Medical/personal
  - Daily standup meetings
- Float is work time, not actual time!
- Velocity can help (pick up the pace) but won’t fix all of it.
- Velocity is NOT a substitute for good estimation; it’s a way of factoring in the real-world performance of you and your team.

PROJECT AWARENESS

- Customer knows where you are
- You know where you are
  - Successful software development entails knowing where you are
  - With an understanding of your progress and challenges, you can keep your customer in the loop, and deliver software when it's needed.
- Q4U
  - Contrast with Big Bang/Waterfall?

TODAY'S CLASS TAKEAWAY

What is the relationship between stories and tasks?

What should we do with our stories before estimating/planning poker?

Why does velocity help us not plan for the unexpected?

NEXT TIME - DESIGN

- Input
  - Requirements
- Output
  - Models and artifacts that record major design decisions