

Project Planning

Requirements Gathering

User Stories

- Brainstorming
- Role playing
- Observation
- Clarifications

Estimates

- Planning poker/swimlane estimation
- Assumptions
- Clarifications
- Velocity

What's next?

Milestone Planning

- The customer has told you what they want
- How important each requirement/feature...is also up to them
 - While you might be able to guess, it's their time and money so they get to decide
- Before you define specific priorities, help the customer identify the features for the first release
 - If you know the customer's time frame for delivery you can help let them know if something can be done in the time available

Priorities

- Once you have all the milestone features assembled, time to decide priorities
- We wait until the feature set is selected to narrow prioritization to a subset of all the user stories
- Customer sets all priorities
- We prioritize for a few reasons:
 - What to do first
 - What if something happens and not everything can be finished...lower priority items might get bumped from the milestone
 - Customer needs to be kept in the loop on this!

Priority Values

- When assigning priority, HFSD recommends starting with 10 as high priority and then using increments of 10 to represent decreasing priority
 - 10, 20, 30, 40, 50
- In most cases, these 5 priorities provide ample granularity, but also gives you a little room if you have a requirement come in that needs slightly higher or lower precedence over other items
- Note that priorities can take ANY other form as long as both the development team and customers understand the system

Now that we know the pool of features the user wants, we can start thinking about implementation.

Tasks

- User stories are for the **customer**
 - Describes a software feature from their perspective
- To the developers, each user story represents a **collection of tasks**
- Tasks are for the developer
 - Describes implementation details to achieve the user story feature
 - Should only be a single development activity for **ONE** developer
- A task has a:
 - Title – Can be a number/identifier or a phrase (something to reference)
 - Description – Short, broad description of the task
 - Estimate – Same process with user stories

Task Estimates

- Ideally, start as early as possible, but might not have time to convert ALL user stories to task right away. In this case you can focus your efforts on user stories in the milestone.
- Tasks are more representative of the work to implement
- Task estimates may not agree with user story
 - Go with the task estimate sum over the user story estimate
- Modern SE practices suggest creating tasks that can be completed in a Day or two (low effort) and HFSD suggests no more than 5 Days (medium effort)
 - Hours are often too short
 - The larger the task the easier it is to lose focus
 - “Create database” might be too big, perhaps “Create customer table” instead.
- Estimate doesn't fit the range? Break it down further!

Milestone 1.0

- Sum up your estimates and see how long it will take with respect to how many work hours you have
 - If using story points, bind iterations to an estimated fixed number of points per iteration
- If the total work is outside the customers release window, reprioritize the feature set to:
 - Cut functionality
 - Ship early to keep momentum
 - Focus on BASELINE functionality
- Other user stories don't have to be discarded, just bumped to a different milestone

Iteration Planning

- Iterations are treated like milestones, but are not the same
 - Iterations are spans of development work that result in a buildable and deliverable unit of work
- Milestones contain the “final” feature set release
- With a set of Milestone 1.0 user stories, we need to place the features into our iteration plan
- Iteration lengths can vary from months to weeks
- Focus on priorities first
 - You might find that time estimates leave gaps that high priority tasks cannot fill

Planning Math

- Number of person-days per iteration:
 - [team size] * [iteration length in days] * [velocity]
 - 3 people * 20-day iteration * 0.7 = 42 person-days
- Total amount of work that can be done for a milestone:
 - [person-days per iteration] * [number of iterations]
 - 42 person-days * 3 iterations = 126 person-days for milestone
- If using story points with a new project/team you will need to make an educated guess at first. After you have data for about three iterations you can average of the work completed over the iterations. Update your iteration plans and make sure you can still complete on time.
 - If not, have a talk with your customer.

Milestone 1

Number of Developers: 5

Iteration Length in Days: 20*

Assumed starting velocity: 0.7

Number of iterations: 4

Person days per iteration:

$$5 * 20 * 0.7 = 70 \text{ person-days}$$

Deal with high priority first, and transition to lower priority as possible. It may be necessary to mix higher and lower priority user stories to fit into an iteration. Note that the total of all user story estimates per iteration must be \leq the number of **Person days per iteration**.

Iteration 1
Person-Days: 69

Title: View an order				Est: 15	Est: 16
Priority: 10				Est: 7	Est: 20
Estimate: 3				Est: 8	Est: 8
T1	T2	...	TN		

Iteration 2
Person-Days: 70

Est: 15	Est: 12	Est: 7
Est: 10	Est: 20	Est: 6

Iteration 3
Person-Days: 68

Est: 15	Est: 18	Est: 5
Est: 14	Est: 10	Est: 6

Iteration 4
Person-Days: 64

Est: 8	Est: 4	Est: 4	Est: 5
Est: 3	Est: 15	Est: 10	Est: 15

Color Key: High Priority Medium Priority Low Priority

*the assumed one-month iteration length by the HFSD book

What About the Customer?

- If the timeline for the milestone and the estimates don't match, they need to know
- Do **NOT** try to crunch and get everything in
 - **Peak** productivity is about 3 hours a day
 - Fatigue will set in and increase the rate of errors and mistakes
 - Ultimately, you'll be lowering the quality of the software anyway and doing so at the cost of your team
- You can add an iteration to make up the time if the customer will wait
- Postpone the work to another milestone release
- Be transparent

Process Recap

- Establish user stories with customer
- Perform user story estimation with development team
 - Can create and estimate tasks here
- Assemble a milestone feature set with the customer
 - Can create and estimate tasks here also (narrower feature set)
 - Update user story estimations if necessary
- Determine user story priorities with the customer
- Assemble a milestone release plan with iterations
 - Revise/re-prioritize milestone feature set if necessary
- Update the iteration plans as work is added/completed

Getting to Work

Getting Organized

- The “Big Board” dedicated to tracking milestone and iteration progress
 - Could be a physical board or wall
 - Could be a digital in software
 - Hybrid approach
- Visual representation of progress
 - What’s in the pipeline, in progress, and completed
 - The burn down chart

The Big Board Anatomy

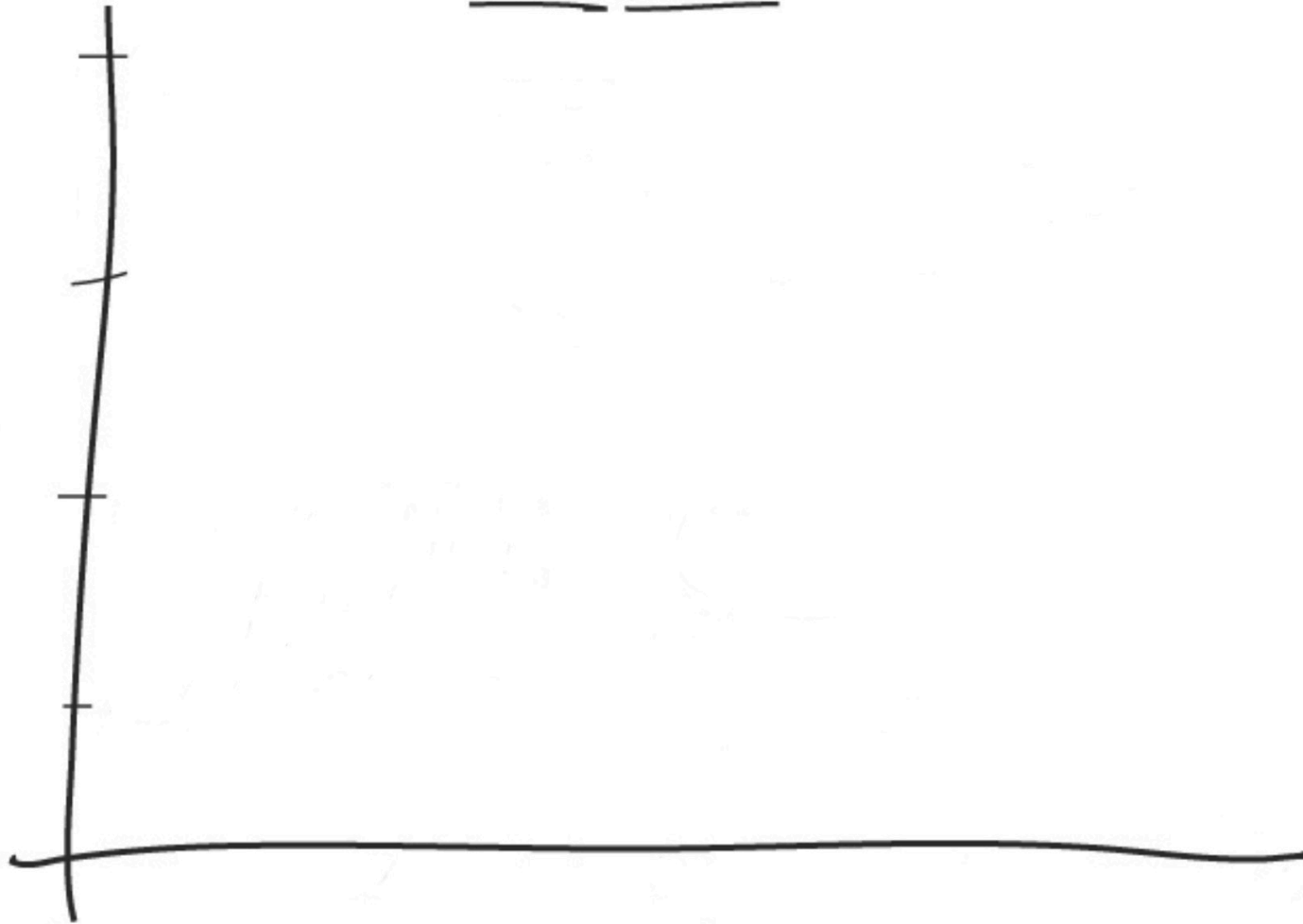
User Stories with Tasks	In-progress	Complete	Burn-down chart
			Next
			Completed
???			

The Big Board Anatomy

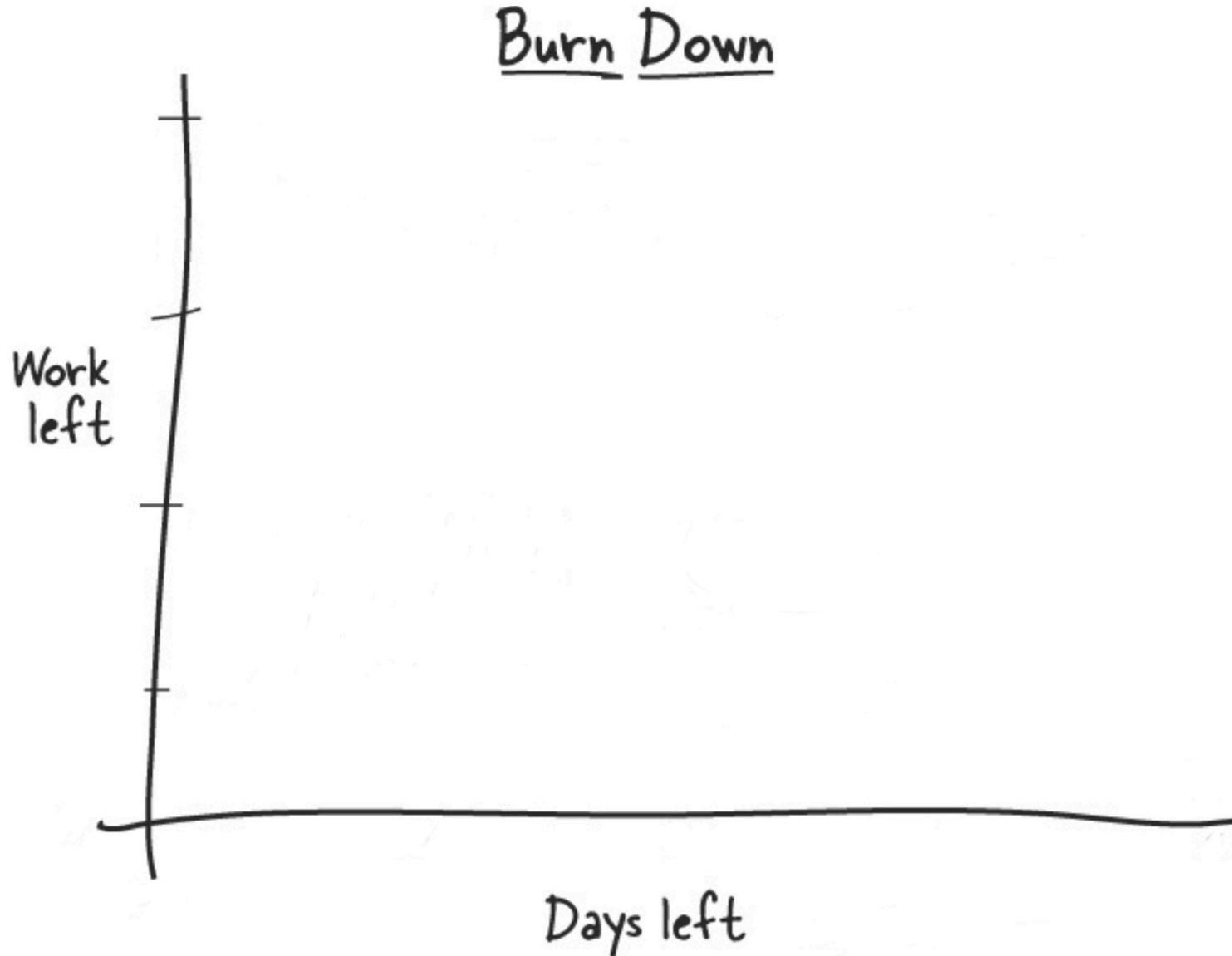
User Stories with Tasks	In-progress	Complete	Burn-down chart
			Next
			Completed
???			

Visualizing Success

Burn Down



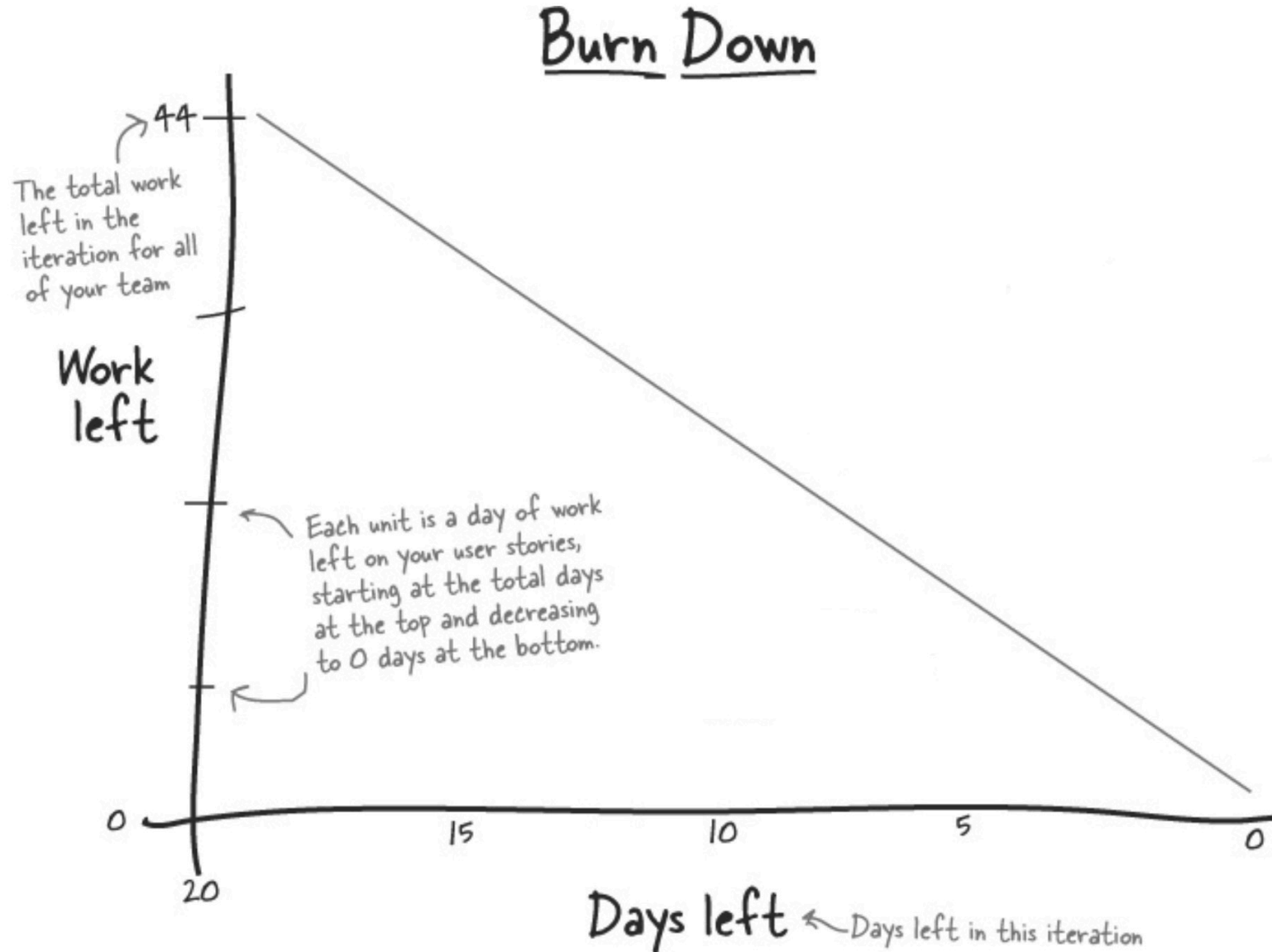
Visualizing Success



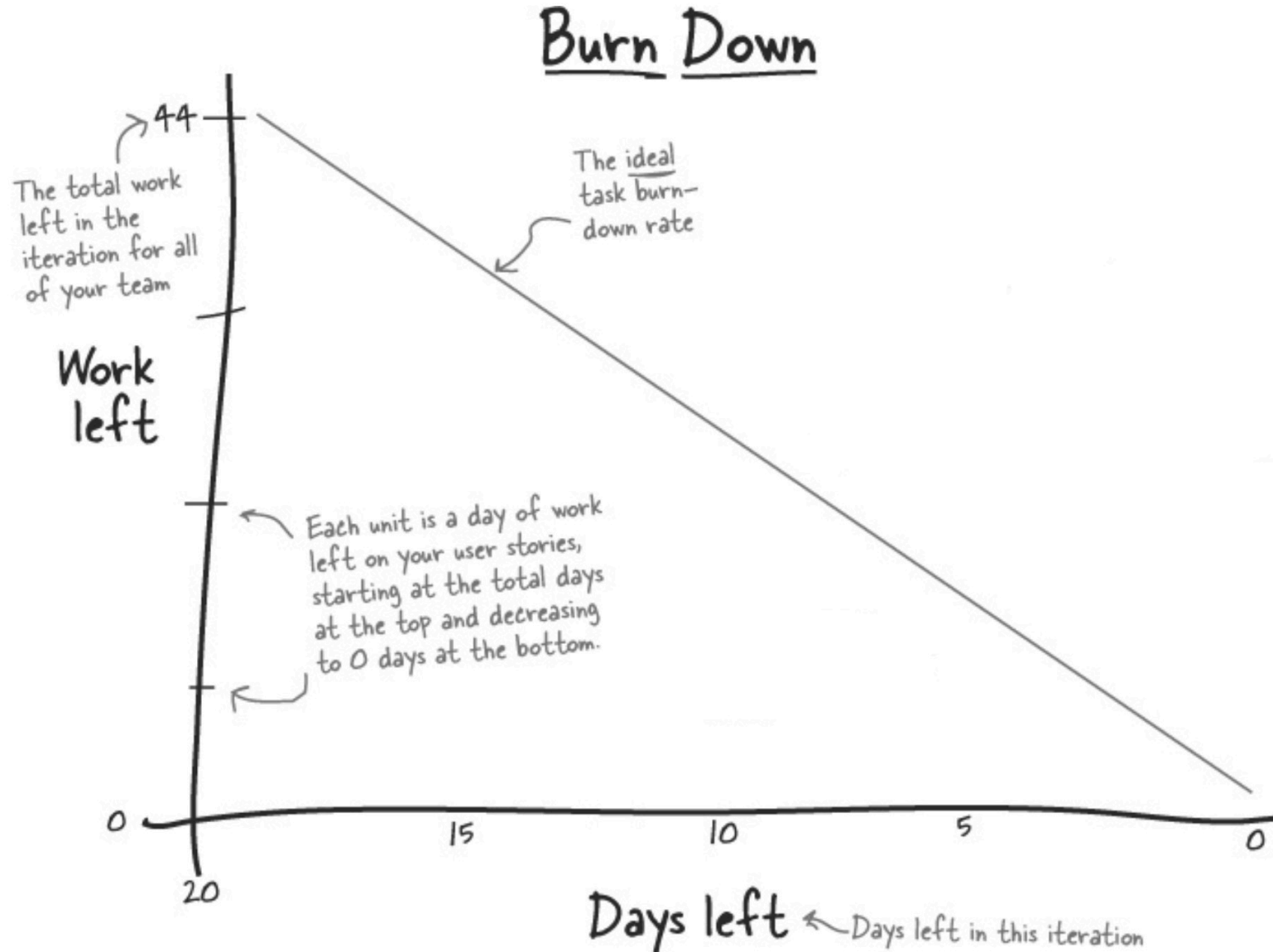
Visualizing Success



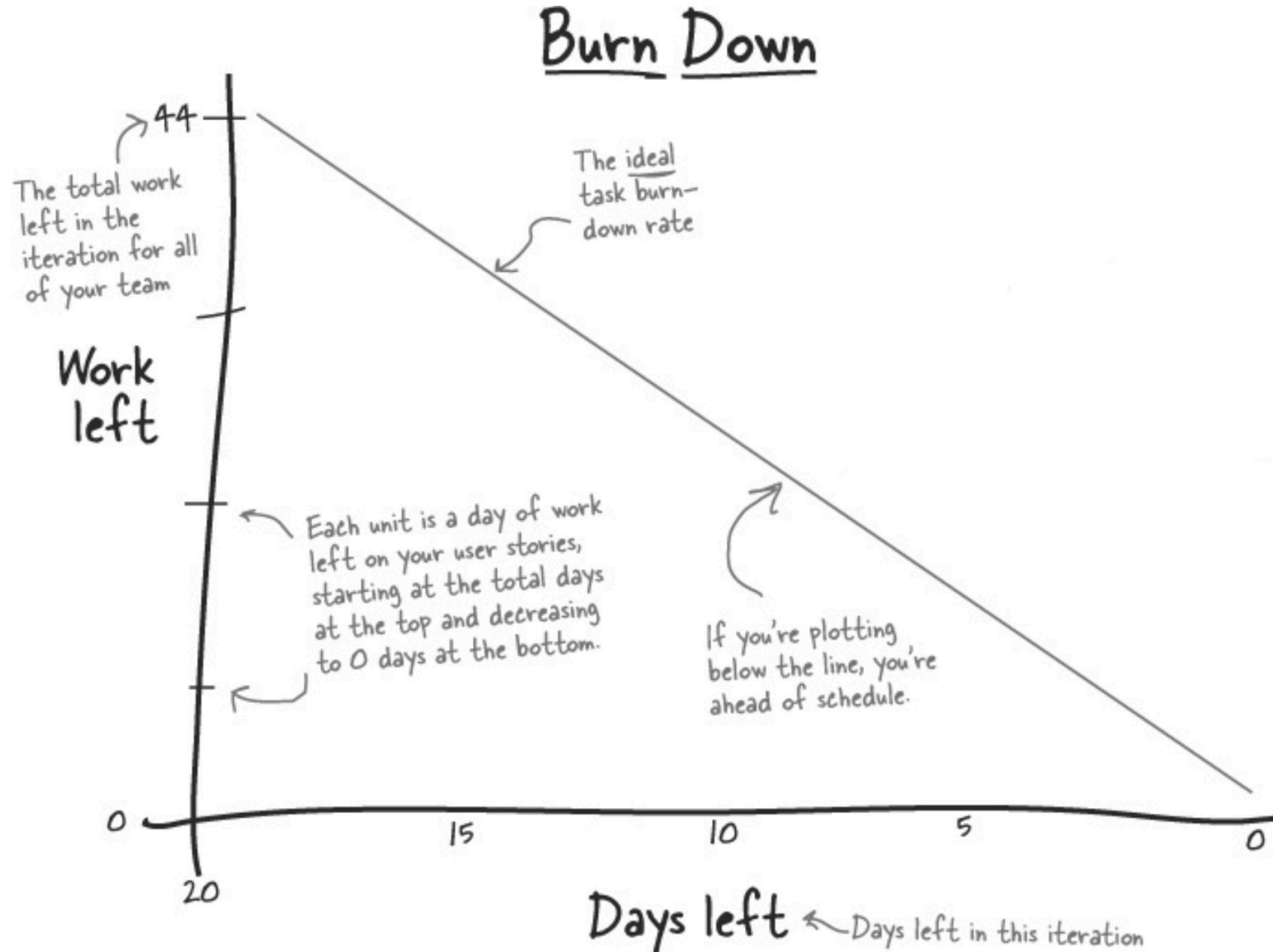
Visualizing Success



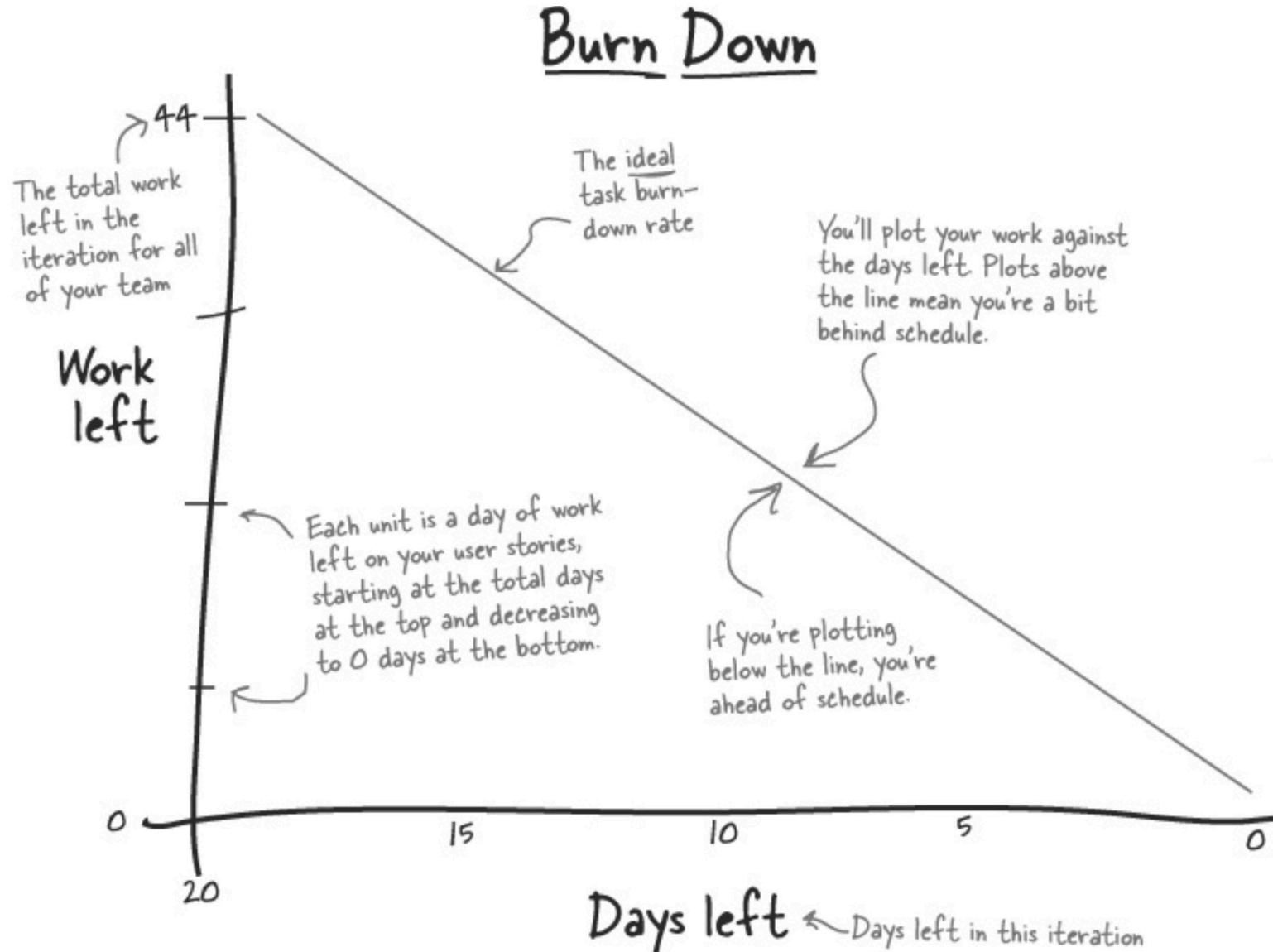
Visualizing Success



Visualizing Success



Visualizing Success



The Big Board Anatomy

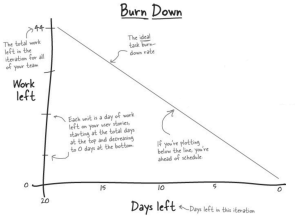

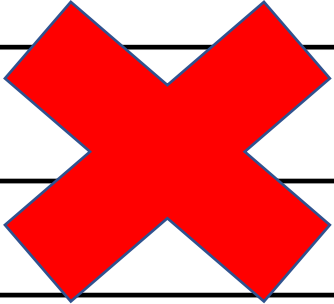


User Stories with Tasks	In-progress	Complete	
			Next
???			Completed

Task Assignment

- Choosing how to split up the work comes down to experience and expertise of the team
 - Use your best judgement
- Don't try to spread everyone out over a bunch of different stories
 - Best to consolidate your efforts to finishing user stories
 - Avoids leaving multiple stories partially completed
- You **CAN** work on two tasks at once, but they need to be reasonably related
- You **CANNOT** take on two tasks with a large estimate
 - Long tasks are difficult enough to keep focus let alone multiple simultaneously
 - Beware of context switching

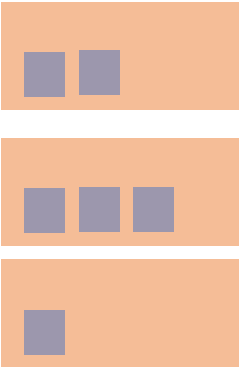
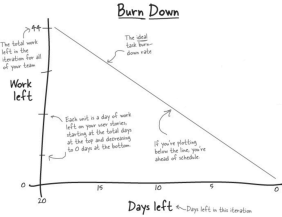



Tasks on the Big Board

- Tasks are NOT in-progress when they are assigned

User Stories with Tasks	In-progress	Complete	
			Next
			
			
???			










Tasks on the Big Board

- When work begins on a task it is moved to “in-progress”
- Moving work to in-progress before work has begun is detrimental to the accuracy of the Big Board

User Stories with Tasks 	In-progress	Complete	
			Next
			
			Completed
???			

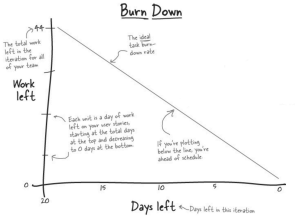






Tasks on the Big Board

- When a task is finished it goes to “complete”

User Stories with Tasks	In-progress	Complete	
			Next
			
			Completed
<p>???</p>			

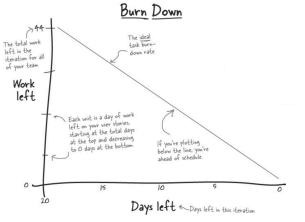






Tasks on the Big Board

- Once all tasks for a user story are done, the user story is moved to “completed”

User Stories with Tasks	In-progress	Complete	 <p>The left work left in the iteration for all of your team The right work left in the iteration for all of your team Each work in a day of work left on your team... dividing all the total story at the top and determining the steps at the bottom If you're finishing before the time you're ahead of schedule</p> <p>Work left</p> <p>Days left ← Days left in this iteration</p>
			Next
			
<p>???</p>			Completed

Tasks on the Big Board

- Once all tasks for a user story are done, the user story is moved to “completed”

User Stories with Tasks	In-progress	Complete	 <p>Burn Down</p> <p>The left work left in the sprint for all of your team</p> <p>The right work left in the sprint for all of your team</p> <p>Work left</p> <p>Each work in a day of work left on your team. Subtracting at the top and decreasing to 0 days at the bottom.</p> <p>If you're finishing before the time you're ahead of schedule.</p> <p>Days left ← Days left in this iteration</p>
			<p>Next</p>
			
			<p>Completed</p> 
<p>???</p>			

Tasks on the Big Board

- Any user stories that are not complete at the end of the iteration go into next so they can be assigned to the next iteration (if necessary)

User Stories with Tasks	In-progress	Complete	
			<p>Next</p>
			<p>Completed</p>
???			

Why Bother?

- Visibility!
- A tangible grasp on the state of a project
 - Are we on ahead, behind, or on target to finish as scheduled
- Conversational Piece
 - A way to have “water cooler” conversations to discuss the project and potential means of improving productivity
 - Some teams switch *AWAY* from purely digital representations to increase the engagement of the team with the work

Metrics and Data Tracking

- All this tracking creates data...we like data.
- Details like...
 - when estimates were off
 - things that negatively impacted the project
 - correlations with slips in a burn down with certain tasks
- Allows for adjustments to the process based on empirical evidence
 - Like changes in velocity
- The process is an evolutionary entity like the software



Meetings...yay...

The Standup Meeting

- Daily (usually morning)
- Intended to be brief such that no one feels the need to sit down
- 5 to 15 minutes
- Focused on short term tasks at hand
- If something in the stand-up meeting requires more attention it is a **separate meeting**

Standup Meeting Activities

- Update the "Big Board" with completed tasks
- Update the burn-down chart
- Talk about what happened yesterday
 - Any issues encountered
 - Accomplishments/successes
- Talk about what will happen today

The Unexpected

- Unplanned tasks will most surely arise
 - Demos
 - Presentations
 - Etc.
- You **MUST** track unplanned tasks as well
- This also counts toward the burn-down rate
- What if this changes the iterations and impacts the milestone?!
 - Talk with the customer and update the iteration based on their priorities

The Big Board Anatomy

User Stories with Tasks	In-progress	Complete	
			Next
Unplanned Tasks			Completed

But we have velocity, that will cover it!



**I'M
A FRAYED
KNOT.**

Velocity and “Float” time

- We added some additional time to our estimates with velocity
- This can help with unexpected issues...but only so much
- Remember that velocity was just an estimate for productive the team might be
- Realistically the “extra” time we received from the velocity is a buffer and can disappear quickly

You Can't Plan for Everything

- You will **NEVER** be able to account for all eventualities
- What you **CAN** do is know where you stand at any given point in time
- While this sounds like a consolation prize, this is important
- Without a process in place to tell you there are problems, development would continue down the wrong path
- Instead, we can adjust by changing our iteration plans, consulting with the customer, and refocus our resources for the best possible outcome