Example: Creating a New Account

- User Story (in Role-Goal-Benefit template):
  - As a new user, I need to create an account so that the server can authenticate potential users

- Questions to Consider in Customer Discussions?

Example: Creating a New Account

- User Story (in Role-Goal-Benefit template):
  - As a new user, I need to create an account so that the server can authenticate potential users

- You might get the customer to tell you that a new account needs a user-name and a password
- If you raise the issue, you might get your customer to tell you something about the password policy
- But most customers will not tell you the details you need to implement this story

Details you Need to Know

- What is the user-name policy?
  - Can new users choose an arbitrary user-name?
  - Must a user-name be a valid eMail address?
  - If it's an eMail address, what happens if their address changes?
- What is the password policy?
  - Minimum length?
  - Contain both upper and lower-case letters?
  - Contain digits?
  - Contain symbols?
- Customers often don't consider these decisions

Exercise: Define Acceptance Criteria for a User Story

- User Story (in Role-Goal-Benefit template) :
- As an existing user, I need to delete my account to help protect my privacy by removing my private data from the server

- Questions to Consider in Customer Discussions?

Questions to Consider in Customer Discussions

- Does deleting an account require any additional security?
  - send user an eMail asking them to confirm deletion?
- What does deleting an account actually delete?
  - User's right to logon the server?
  - User's data on the server?
- If deleting an account doesn't delete the user's data, what happens if the user later decides to recreate the account?
- Note: Some questions that arise when discussing Acceptance Criteria might discover new user-stories

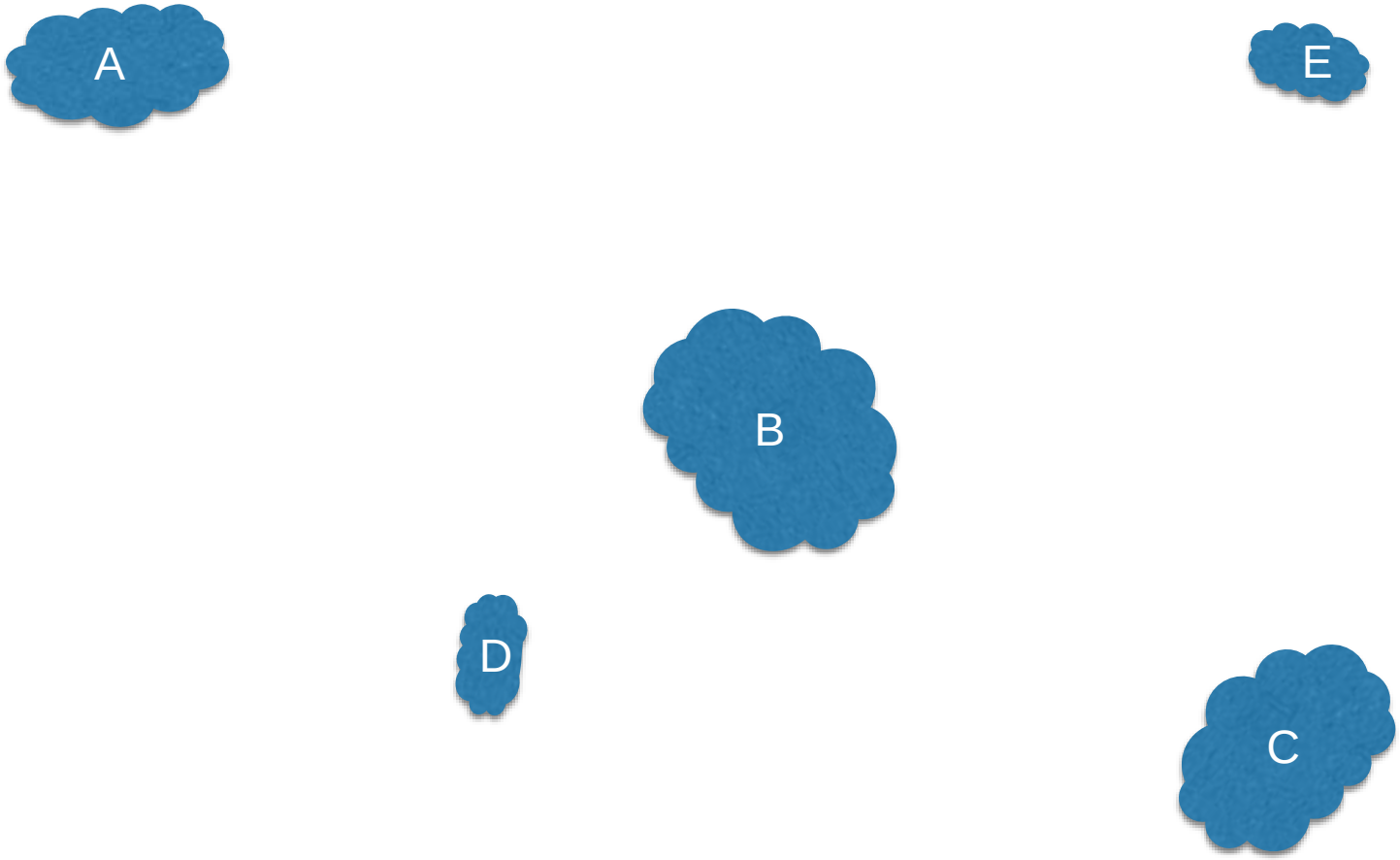
Be Pro-Active when Gathering Requirements!

- Note that the customers just aren't going to tell you everything you need to know

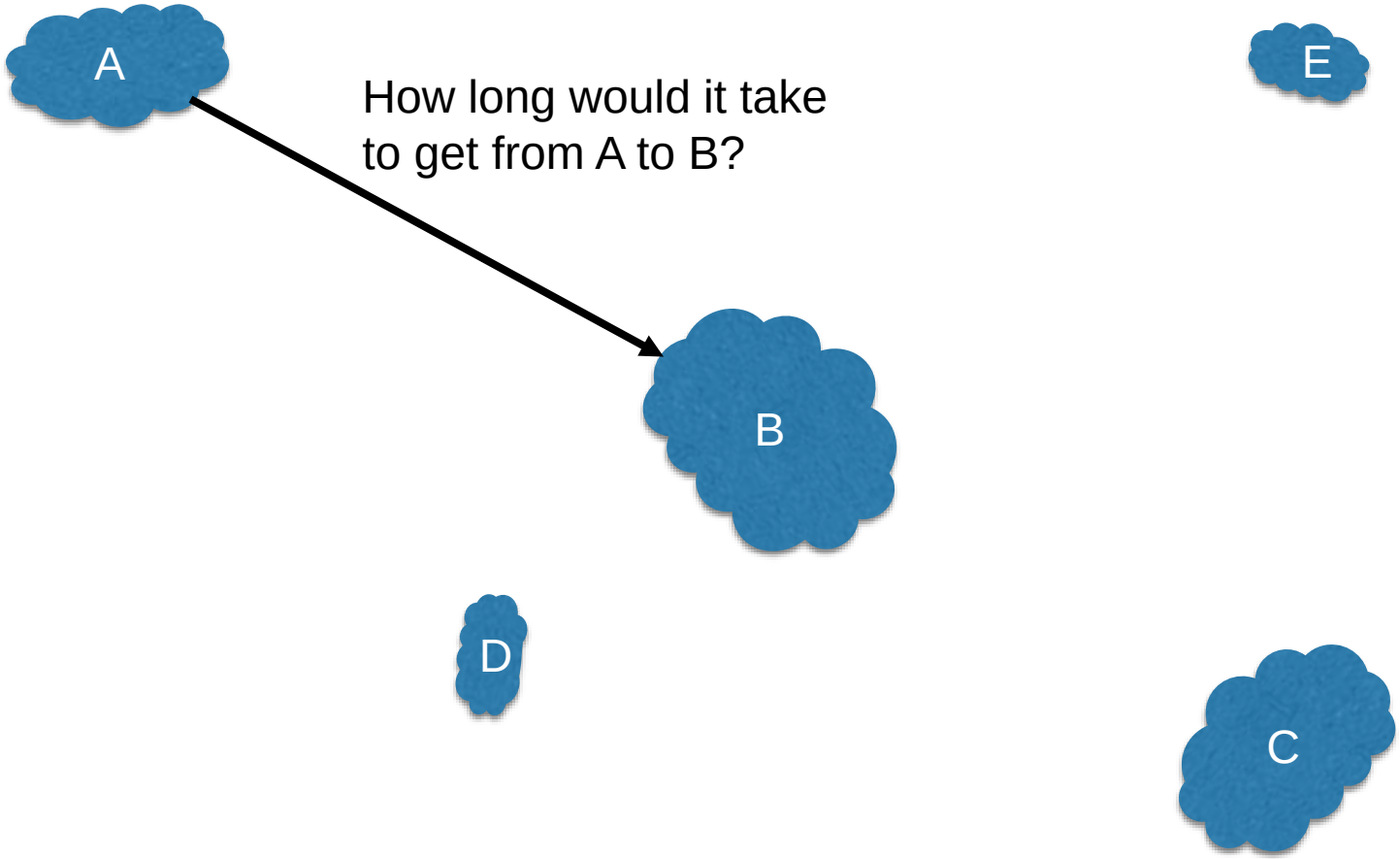
- You have to engage them in discussions

Estimating In Scrum

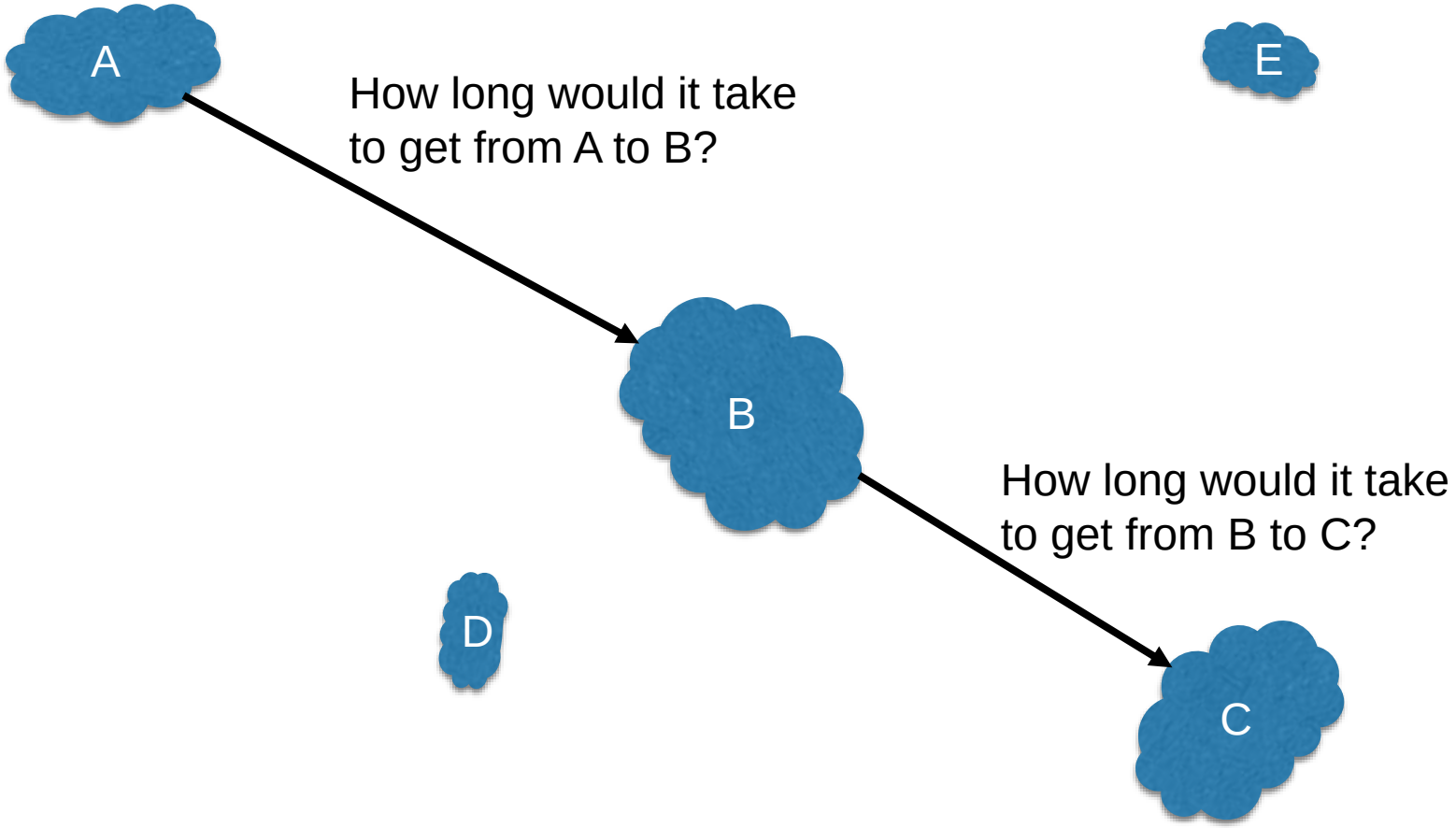
Quick Estimation Exercise - Archipelago



Quick Estimation Exercise - Archipelago



Quick Estimation Exercise - Archipelago



Why Estimate?

Why Estimate?

- Business makes trade-offs
  - Market segments (of users in different roles)
  - Features (user stories)
  - Expected revenue
  - Schedule (product availability)
  - Quality
  - Resources
- "When-it-will-be-done" impacts business choices
- Good estimates help business make good choices and decisions (i.e., create more value/$$$)!

Estimation Methods

Estimation Methods

- Guess: Your own wild guess
- Delphi method: The average of everyone's guess
- Experience:
  - Completed task X required N hours to implement
  - Task Y is similar to X only easier (should take $\leq N$ hours to implement)
- Quantitative

Quantitative Estimation Methods

- Example 1:
  - You're asking us to implement 50 stories
  - historically we implement an average of 1 story/engineerDay
  - this will require … engineerDays

Quantitative Estimation Methods

- Example 1:
  - You're asking us to implement 50 stories
  - historically we implement an average of 1 story/engineerDay
  - this will require 50 engineerDays

Quantitative Estimation Methods

- Example 2:
  - You're asking us to implement 1 story having 5 Tasks which require 2, 4, 9, 7 and 3 hours each.
  - Therefore, we'll need … hours

Quantitative Estimation Methods

- Example 2:
  - You're asking us to implement 1 story having 5 Tasks which require 2, 4, 9, 7 and 3 hours each.
  - Therefore, we'll need 2+4+9+7+3=25 hours

Quantitative Estimation Methods

- Example 3:
- X is a bit more difficult than the similar open-source Y which contains 30,000 LOC.
  - We produce an average of 30 LOC/engineerDay so
  - we'll need at least … engineerDays if we write it all ourselves.

Quantitative Estimation Methods

- Example 3:
- X is a bit more difficult than the similar open-source Y which contains 30,000 LOC.
  - We produce an average of 30 LOC/engineerDay so
  - we'll need at least 1000 engineerDays if we write it all ourselves.

Quantitative Estimation Methods

- Example 3:
- X is a bit more difficult than the similar open-source Y which contains 30,000 LOC.
  - We produce an average of 30 LOC/engineerDay so
  - we'll need at least 1000 engineerDays if we write it all ourselves.
    - Q: what are some implications of writing an application similar to an OSS project vs. using an OSS project?

Quantitative Estimation Methods

- Example 3:
- X is a bit more difficult than the similar open-source Y which contains 30,000 LOC.
  - We produce an average of 30 LOC/engineerDay so
  - we'll need at least 1000 engineerDays if we write it all ourselves.
    - Q: what are some implications of writing an application similar to an OSS project vs. using an OSS project?
    - Q: Difference between having:
      - a team of 1000 engineers writing 30LOC/engineerDay
        - will they complete the project in 1 day?
      - a team of 10 engineers writing 30LOC/engineerDay
        - will they complete the project in 100 days?

Estimating Exercise

Exact Estimates:
How tall (in meters) is this building?

Relative Estimates: Which building is taller?

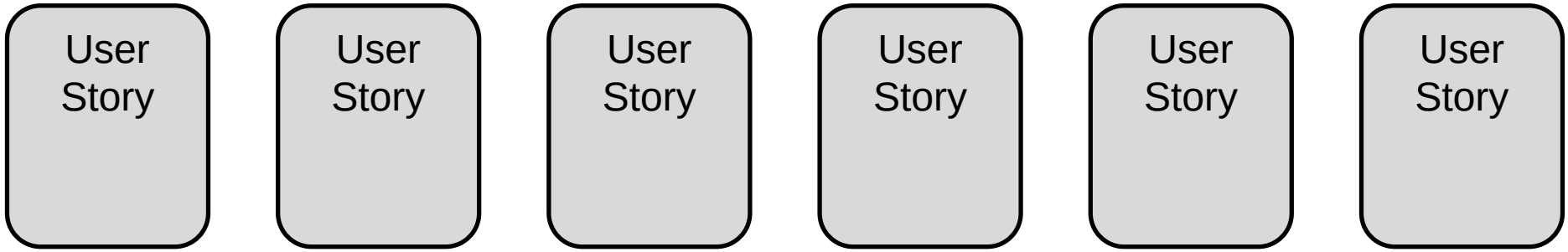
Relative Estimates: Which building is taller?

- and by how much?

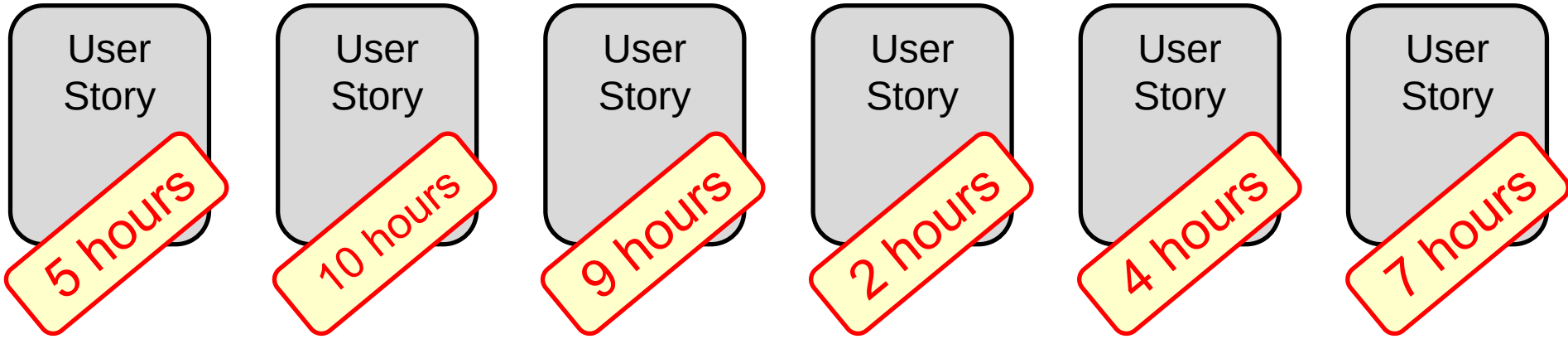
Relative Estimates: Which building is taller?

- and by how much?

How Would Developers Estimate the User Stories in the Product Backlog?

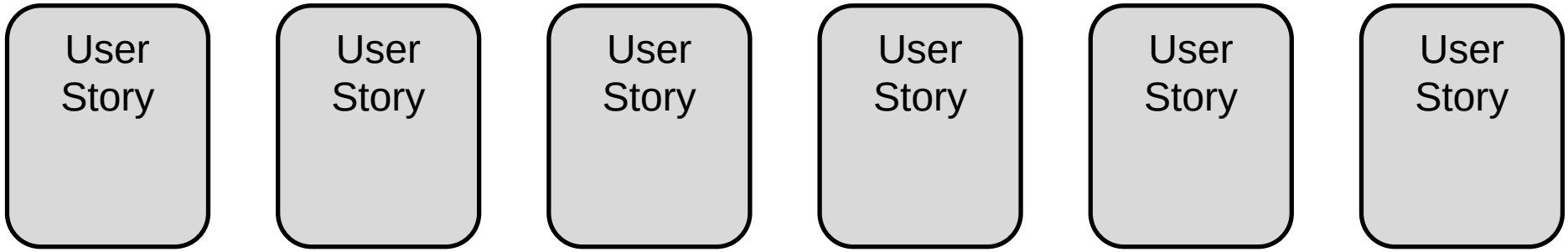


How Would Developers Estimate the User Stories in the Product Backlog?



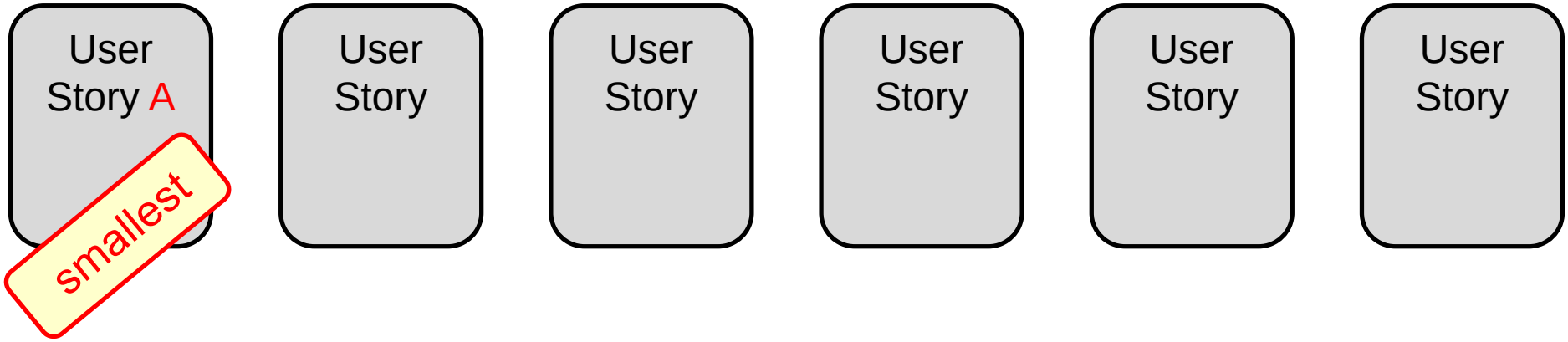
- Using an absolute size (i.e., hours) or each story

How Would Developers Estimate the User Stories in the Product Backlog?



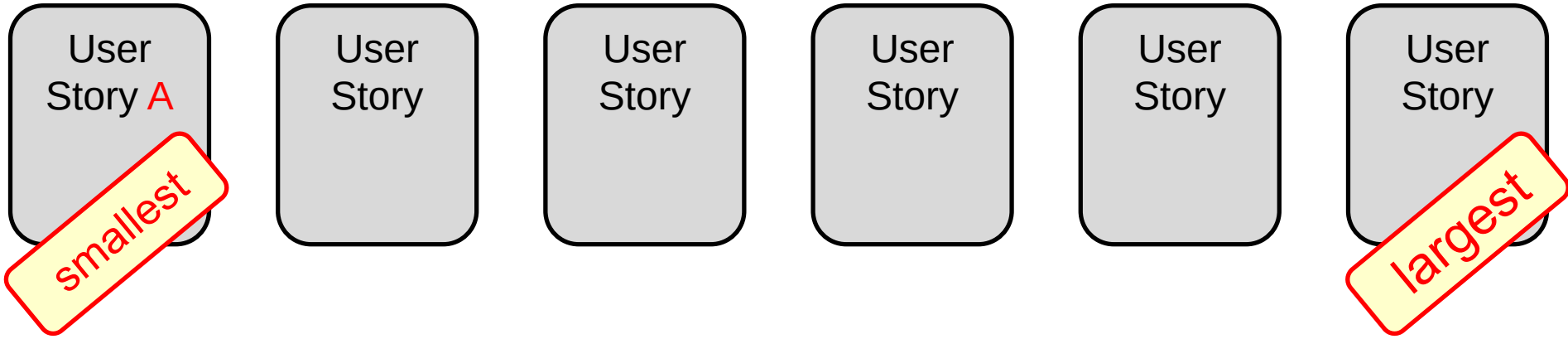
- What would be an alternative approach?

How Would Developers Estimate the User Stories in the Product Backlog?



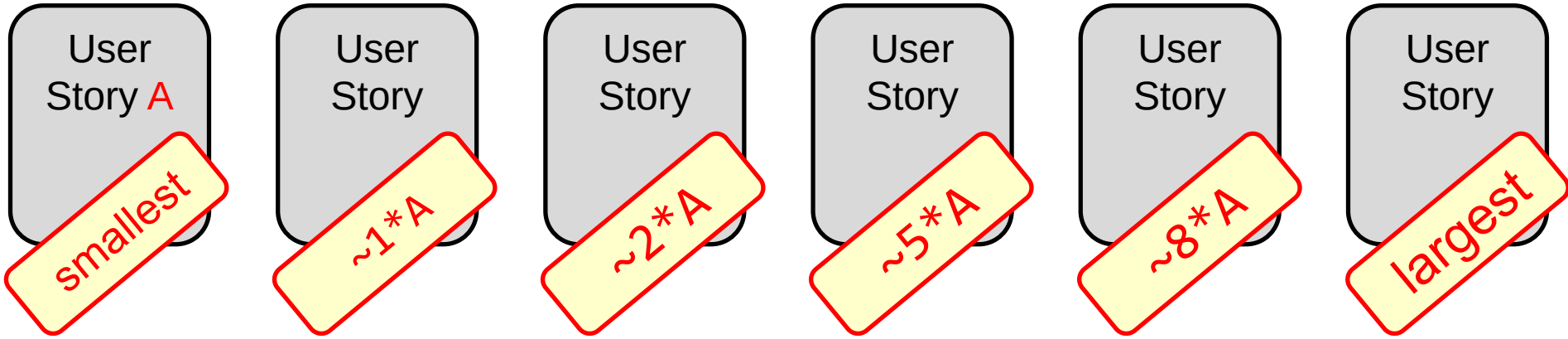
- Identify the "smallest" story

How Would Developers Estimate the User Stories in the Product Backlog?



- Identify the "smallest" story
- or
- Sort the user stories from "smallest" to "largest"

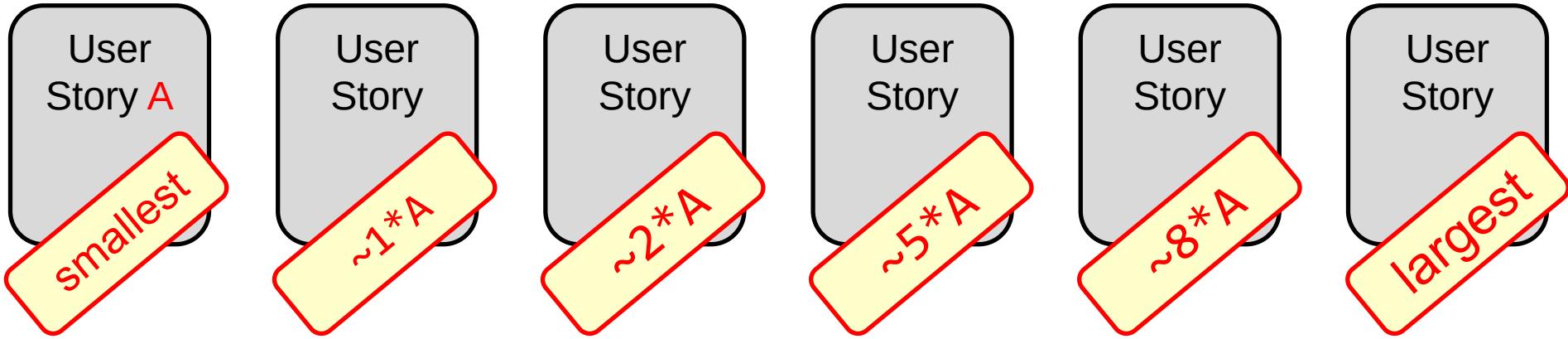
How Would Developers Estimate the User Stories in the Product Backlog?



- Identify the "smallest" story
- or
- Sort the user stories from "smallest" to "largest"

- Estimate the other stories relatively to smallest story

How Would Developers Estimate the User Stories in the Product Backlog?



- Replace "smallest" story A as a unit of measure with a generic unit: story point
- story point = relative unit of measure for the amount of work needed to complete a user story

Exercise: What is the Difference Between

- a 53-story point vs. a 57-story point

- a 1-story point vs. a 5-story point

Exercise: What is the Difference Between

- a 53-story point vs. a 57-story point
  - not important
- a 1-story point vs. a 5-story point
  - Important

- Many scrum teams use Fibonacci numbers
  - Allows accurate estimate smaller user stories
  - Allows coarser estimates for larger user stories

Estimating User Stories with Planning Poker

- AKA Scrum Poker
- Wisdom of the Crowds (Delphi) approach to estimation
- Primarily used during Initial Planning (prior to the first sprint) and Backlog Grooming

Estimating User Stories with Planning Poker

- Scrum Master moderates but does not play
- Product Owner explains the story, answering questions from team

Estimating User Stories with Planning Poker

- Each Developer has a set of cards representing estimates (story points), e.g.,:
- $0, \frac{1}{2}, 1, 2, 3, 5, 8, 13, 20, 40, 100$ (or ∞), and ?

"analog" / paper or digital cards

http://www.softwaretestinghelp.com/planning-poker-scrum-poker-cards-agile-estimation/

https://www.planningpoker.com/

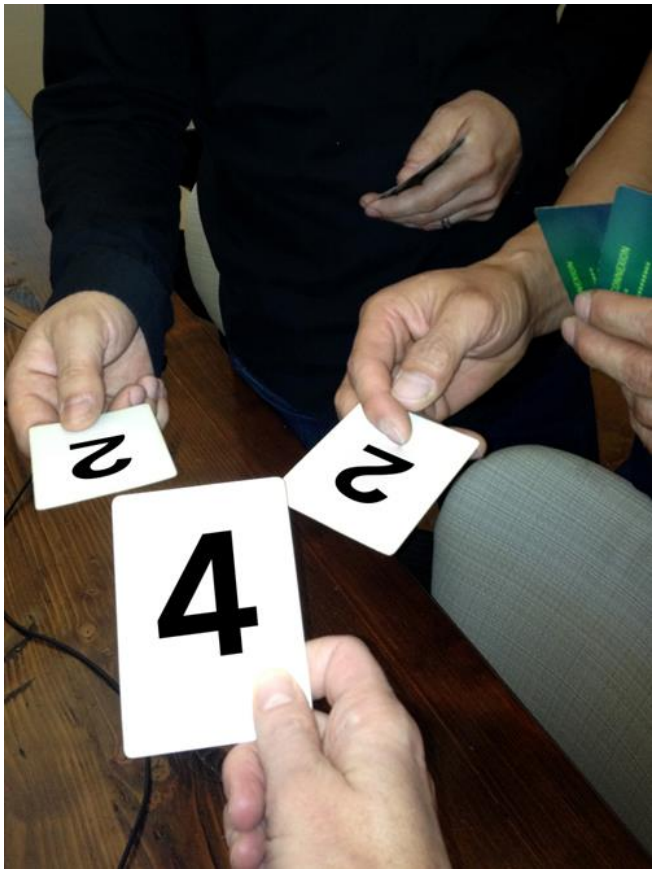
Estimating User Stories with Planning Poker

- Each Developer chooses and plays a card representing their estimate (in hours or story points) for the current story

https://www.mountaingoatsoftware.com/blog/3-roles-that-need-to-be-involved-in-agile-estimating-with-planning-poker

Estimating User Stories with Planning Poker

- Everyone simultaneously turns-over their card



http://www.museumsandtheweb.com/mw2012/papers/agile_games_for_productive_teams_0.html

http://www.aubryconseil.com/post/Utilisation-des-cartes-de-planning-poker

Planning Poker Examples:

- A user story reveals:
  - 3, 8, 5, 2

Planning Poker Examples:

- A user story reveals:
  - 3, 8, 5, 2
    - Developers with cards 2 and 8 (i.e., the extreme ranges) make their point and team re-votes

Planning Poker Examples:

- A user story reveals:
  - 3, 8, 5, 2
    - Developers with cards 2 and 8 (i.e., the extreme ranges) make their point and team re-votes
- A user story reveals:
  - 8, 13, ?, ?

Planning Poker Examples:

- A user story reveals:
  - 3, 8, 5, 2
    - Developers with cards 2 and 8 (i.e., the extreme ranges) make their point and team re-votes

- A user story reveals:
  - 8, 13, ?, ?
    - There could be Product or Technology uncertainty ⇒ Discussion

Estimating User Stories with Planning Poker

- High and Low estimates explain their reasoning
- Play again until a consensus or timeout is reached
- If consensus is unobtainable, then write a task to investigate

Estimating the Amount of Work that can be Done in a Sprint: Velocity

Estimating the Amount of Work that can be Done in a Sprint: Velocity

- Teams assigns relative sizes to their stories using story points as the unit
- Team does a couple of sprints and measure how many stories they can implement/fit in the Sprint ⇒
  - velocity = average number of story points per sprint

Estimating Tasks and Epics

- Estimates for epics are especially crude!
  - Used to make broad business decisions
  - Too crude for Sprint Planning
- Break epics into smaller stories for Sprint Planning

- Estimates for tasks should be done by the developer who volunteers for the task!

Relative Sizes vs. Time Estimates

- human beings are:
  - really bad at estimate how long things will take
    - i.e., bad at absolute sizing

2015 CHAOS report from Standish Group

CHAOS RESOLUTION BY AGILE VERSUS WATERFALL

SIZE	METHOD	SUCCESSFUL	CHALLENGED	FAILED
All Size Projects	Agile	39%	52%	9%
	Waterfall	11%	60%	29%
Large Size Projects	Agile	18%	59%	23%
	Waterfall	3%	55%	42%
Medium Size Projects	Agile	27%	62%	11%
	Waterfall	7%	68%	25%
Small Size Projects	Agile	58%	38%	4%
	Waterfall	44%	45%	11%

The resolution of all software projects from FY2011–2015 within the new CHAOS database, segmented by the agile process and waterfall method. The total number of software projects is over 10,000.

https://www.infoq.com/articles/standish-chaos-2015

Relative Sizes vs. Time Estimates

- human beings are:
  - really bad at estimate how long things will take
    - i.e., bad at absolute sizing
  - really good at estimating relative sizes
    - i.e., good at relative sizing

Relative Sizes vs. Time Estimates

- human beings are:
  - really bad at estimate how long things will take
    - i.e., bad at absolute sizing
  - really good at estimating relative sizes
    - i.e., good at relative sizing

- We are good at comparing two things A and B and
  - judging which one is bigger/smaller
  - how many times A is bigger/smaller than B

Applications of Scrum

(Williams_ESEM2011_Scrum_3_Microsoft_Teams.pdf)
on Piazza