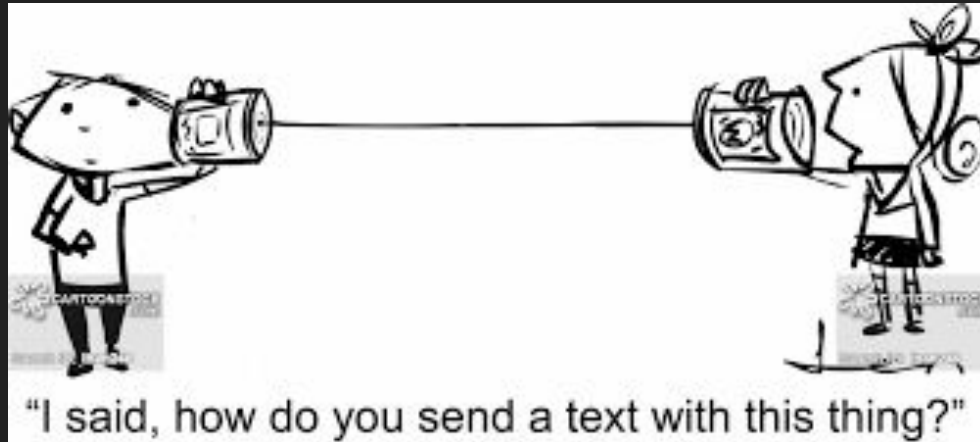


User Stories

Adapted from slides by Mike Cohn

What Problem Do Stories Address?

- Software requirements are a communication problem
- Those who want the software have to communicate with those who will build it



Balance is Critical

Business interests

- Functionality defined by ambiguous natural languages
- Unrealistic timelines are set
- Features are set without understanding of how realistic they are

Technical interests

- Functionality defined by technical jargon
- Well documented code
- Fully tested code
- Feature bloat
- developers lose the opportunity to learn from listening



Resource Allocation

- Projects fail when the problem resources are poorly allocated
- Communication is necessary so that resource allocation becomes a shared problem
- Business team and technical team speak different languages
- Need a method to communicate so both sides can agree on resource allocation, leading to project success!

Responsibility for Resource Allocation

- If developers are responsible
 - May trade quality for additional features
 - May only partially implement a feature
 - May solely make decisions that should involve the business
- If the business is responsible
 - Lengthy upfront requirements negotiation and signoff
 - Features are progressively dropped as the deadline nears

Imperfect Schedules

- We cannot perfectly predict a software schedule
 - As users see the software, they come up with new ideas
 - Too many intangibles
 - Developers have a notoriously hard time estimating
- If we can't perfectly predict a schedule, we can't perfectly say what will be delivered

So what do we do?

- We make decisions based on the information we have, but we do it often
- Rather than making one all-encompassing set of decisions, we spread decision making across the project
- This is where **user stories** come in

What Are **User Stories**?

The Three C's

- Card
 - Stories are traditionally written on note cards
 - Cards may be annotated with estimates, notes, etc
- Conversation
 - Details behind the story come out during conversations with product owner
- Confirmation
 - Acceptance tests confirm a story was coded correctly

Samples from a Travel Website

As a user, I want to reserve a hotel room.

As a vacation traveler, I want to see photos of the hotels.

As a user, I want to cancel a reservation.

As a frequent flyer, I want to rebook a past trip so that I save time booking trips I take often.

Where are the details?

- As a user, I can cancel a reservation
 - Does the user get a full or partial refund?
 - Is the refund to her credit card or is it site credit?
 - How far ahead must the reservation be cancelled?
 - Is that the same for all hotels?
 - For all site visitors? Can frequent travelers cancel later?
 - Is a confirmation provided to the user?
 - How?

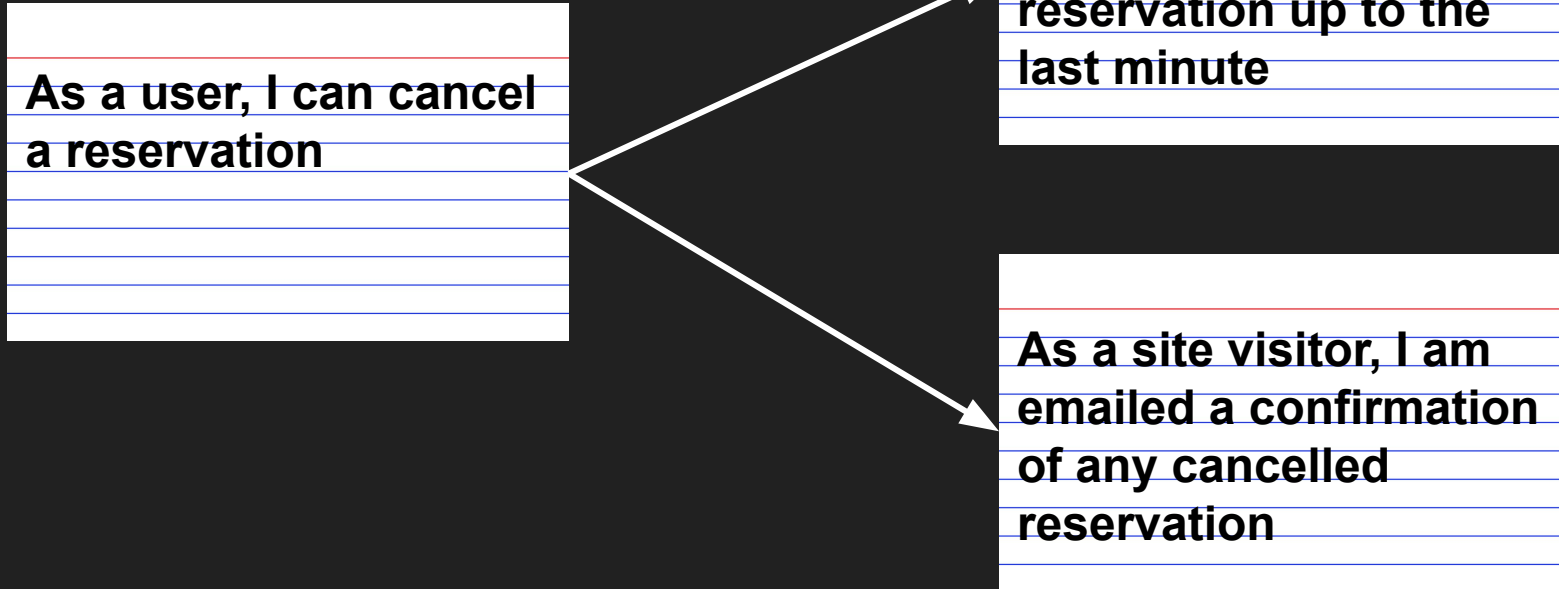
Details as Conditions of Satisfaction

As a user, I can
cancel a reservation.

- The product owner's conditions of satisfaction can be added to a story
- These are essentially tests

- Verify that a premium member can cancel the same day without a fee.
- Verify that a non-premium member is charged 10% for a same-day cancellation.
- Verify that an email confirmation is sent.
- Verify that the hotel is notified of any cancellation.

Details added in Smaller Sub-stories



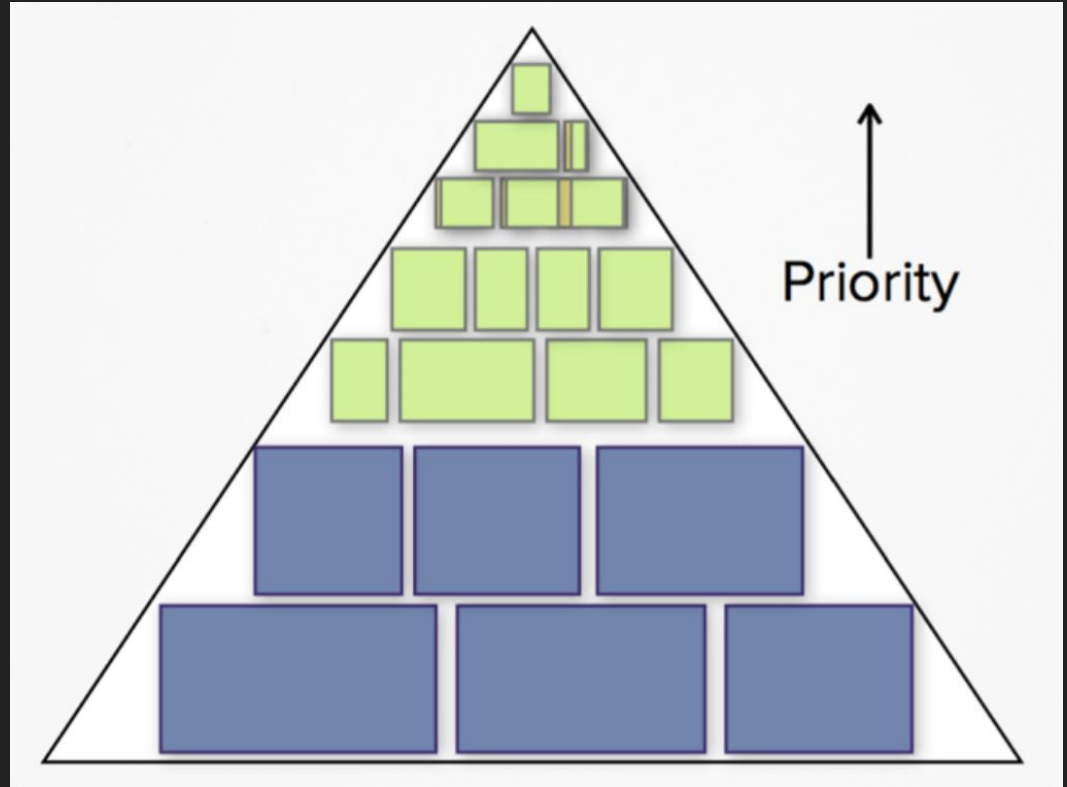
Techniques can be Combined

- These approaches are not mutually exclusive
- Write stories at an appropriate level
- By the time it's implemented, each story will have conditions of satisfaction associated with it

The Product Backlog Iceberg

Theme: a collection of related user stories

Epic: a large user story



An Example

As a VP Marketing, I want to review the performance of historical promotional campaigns so that I can identify and repeat profitable ones.

Clearly an epic

As a VP Marketing, I want to select the timeframe to use when reviewing the performance of past promotional campaigns, so that ...

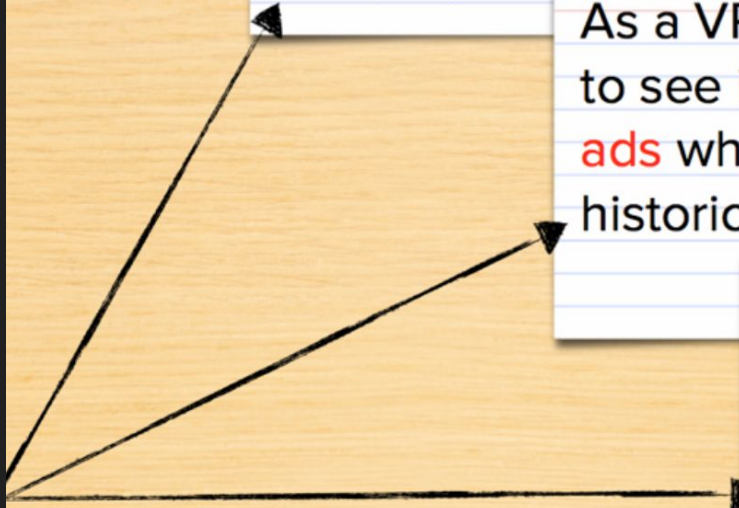
Epics???

As a VP Marketing, I can select which type of campaigns (direct mail, TV, email, radio, etc.) to include when reviewing the performance of past ...

As a VP Marketing, I want to see information on **direct mailings** when reviewing historical campaigns.

As a VP Marketing, I want to see information on **TV ads** when reviewing historical campaigns.

As a VP Marketing, I want to see information on **email ads** when reviewing historical campaigns.



Writing User Stories

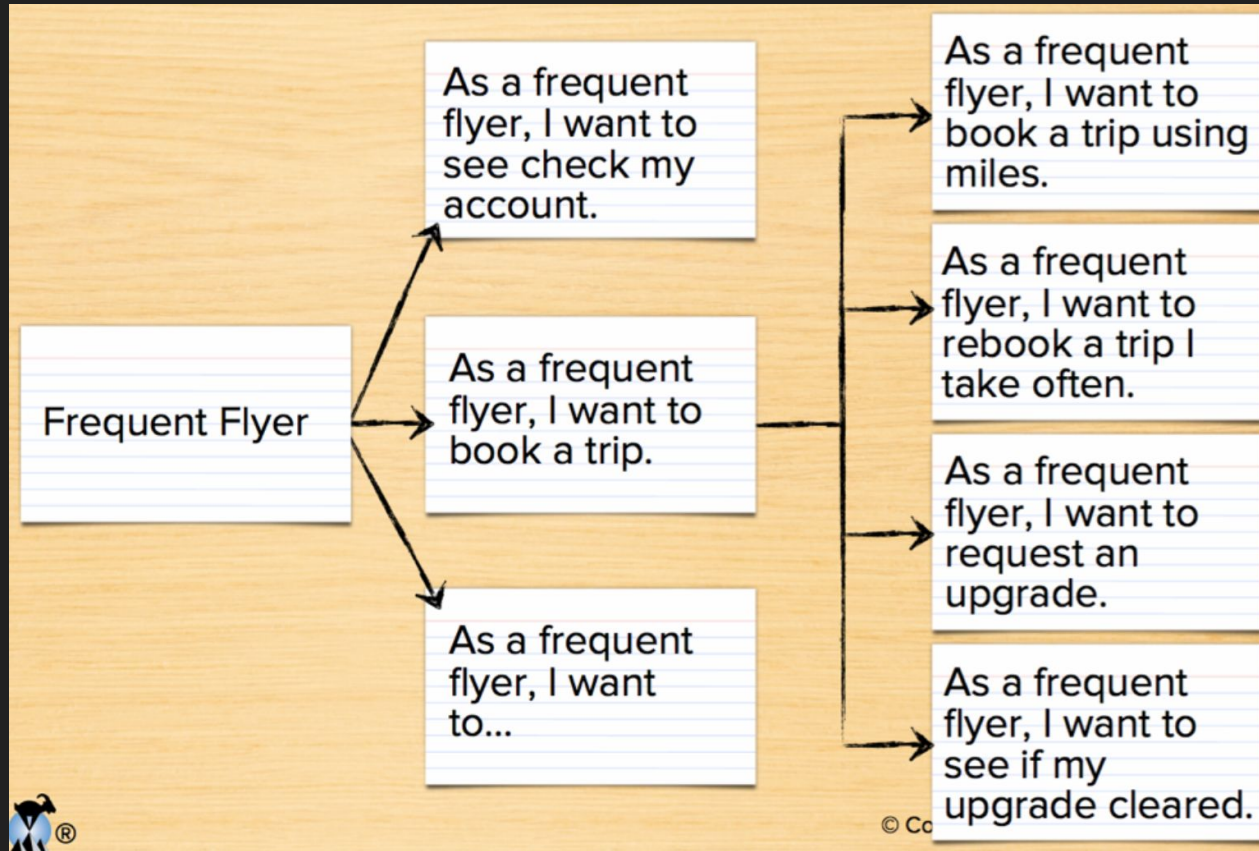
Exercise - Logging In

- See how many user stories you can write about logging in
- Examples:
 - As a registered user, I am required to login so that I can access the system
 - As a forgetful user, I can request a password reminder so that I can log in if I forget mine
- “As a <user role>, I <want/need/can/etc> <goal> so that <reason>”

Story-writing Workshop

- Includes whole team plus possibly some external stakeholders
- Typically not done every sprint
- Brainstorm to generate stories
- Goal is to write as many stories as possible
 - Some will be “implementation ready”
 - Others will be epics
- No prioritization at this point

Start with Epics and Iterate



Why User Stories?

So why User Stories?

- Shift focus from writing to talking
 - If requirements are written down, at best you'll get what was written
 - "You built what I asked for, but it's not what I need"
- Words are imprecise
 - "Entree comes with soup or salad and bread"
 - (Soup or Salad) and Bread
 - (Soup) or (Salad and Bread)

Examples

- “The user can enter a name. It can be 127 characters”
 - Must the user enter a name?
 - Can it be other than 127 characters?
- “The system should prominently display a warning message whenever the user enters invalid data”
 - What does “should” mean?
 - What does prominently display mean?
 - Is invalid data defined elsewhere?

Additional Reasons

- Stories are understandable
 - Developers and customers understand them
 - People are better able to remember events if they are organized into stories
- Support and encourage iterative development
 - Can easily start with epics and disaggregate closer to development time

Yet more Reasons

- Stories are the right size for planning
- Stories support opportunistic development
 - We design solutions by moving opportunistically between top-down and bottom-up approaches
- Stories support participatory design

Most Importantly

- Don't forget the purpose
 - The story text we write on cards is less important than the conversations we have