Chapter 4.5 - User Story

MI Qing (Lecturer)

Telephone: 15210503242

Email: miqing@bjut.edu.cn

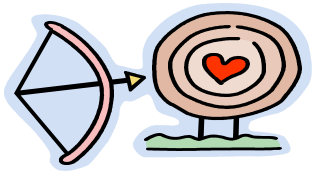
Office: 410, Information Building

Sprint Planning Meeting

Product Backlog

Sprint Planning Meeting
Part One

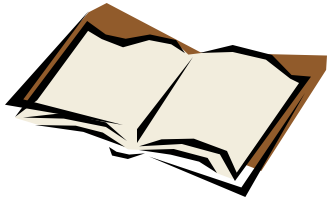
What?



Sprint Goal

How?

Sprint Planning Meeting
Part Two



Sprint Backlog

User Stories for Requirements

- The most common way of writing a product backlog is through user stories.
- The customer chooses the stories for inclusion in the next release based on their priorities and the schedule estimates.

User Story #7	*View Shopping Cart Contents*
As a registered customer, I want to be able to view all items currently in my Shopping Cart together with their prices, so that I can see how much money I am spending.	
Priority:	Estimate:

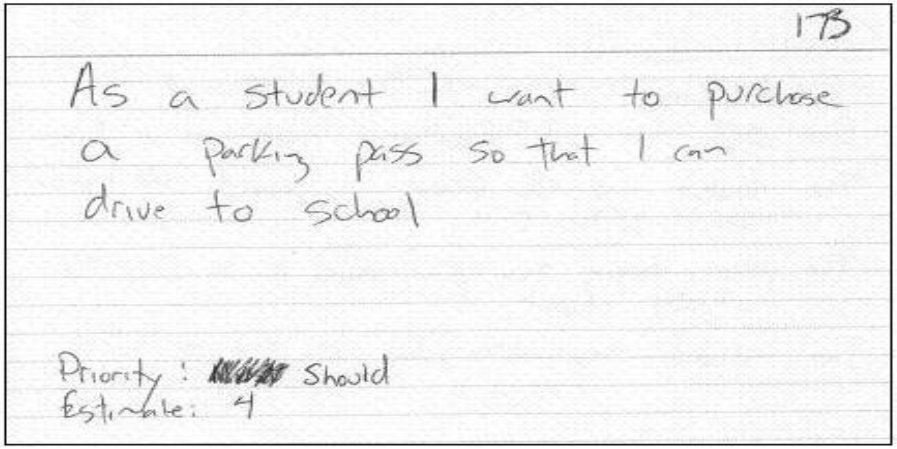
What is a User Story?

- A user story describes functionality that will be valuable to either a user or purchaser of a system or software.
- User stories are composed of three aspects:
  - Card: a written description of the story used for planning and as a reminder
  - Conversation: conversations about the story that serve to flesh out the details of the story
  - Confirmation: tests that convey and document details and that can be used to determine when a story is complete
- While the Card may contain the text of the story, the details are worked out in the Conversation and recorded in the Confirmation.

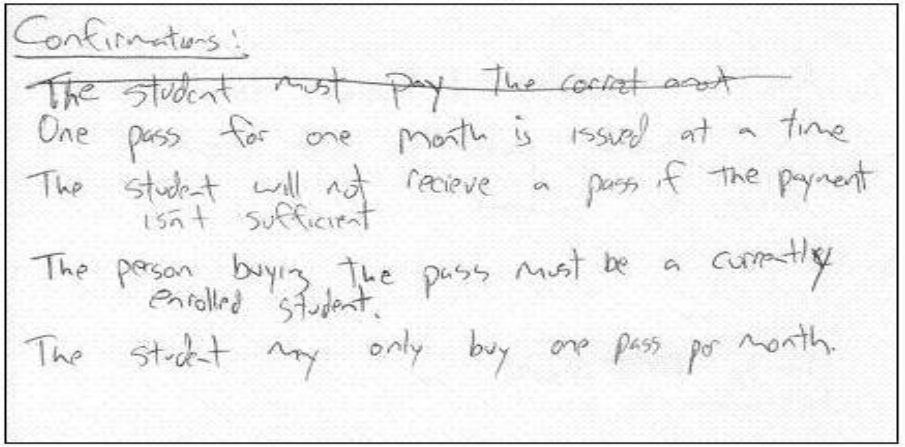
Card

- ✧ Usually written on a plain card, this is to give a physical constraint which limits the possible length of the story.
- ✧ As a <role>, I want <some goal> so that <some reason>.

Front of Card



Back of Card



Card Example

- Here's several hypothetical examples written for YouTube. I've defined a "Creator" as someone who contributes videos to the site, and "User" as someone who just watches them:
  - As a Creator, I want to upload a video so that any users can view it.
  - As a User, I want to search by keyword to find videos that are relevant to me.

Conversation

- Think of the conversation as an open dialogue between everyone working on the project, and the client.
- Anyone can raise questions, ask for things to be clarified, and the answers can be recorded down as bullet points for later reference.
- It is also a stage where you can reevaluate your user story, and possibly split it into multiple stories if required.

Conversation Example

- For example, we discuss the creator upload user story, and decide that there are actually multiple things that can happen, so we split them, and expand on them.

As a Creator, I want to upload a video so that any users can view it.

- The “Upload” button will be a persistent item on every page of the site.
- Videos must not be larger than 100MB, or more than 10 minutes long.
- File formats can include .flv, .mov, .mp4, .avi, and .mpg.
- Upload progress will be shown in real time.

Confirmation

- The confirmation is basically just a test case. If you're not familiar with test cases and test plans, think of a test case as a series of steps that a user must do to achieve the User Story. A test plan is a collection of test cases.
- With full coverage of your system in your test plans, you can easily test every core piece of functionality and tick them off as you go through them.

Confirmation Example

- ✧ As a Creator, I want to upload a video so that any users can view it.
  - Click the “Upload” button.
  - Specify a video file to upload.
    - Check that .flv, .mov, .mp4, .avi, and .mpg extensions are supported.
    - Check that other filetypes aren’t able to be uploaded.
    - Check that files larger than 100MB results in an error.
    - Check that movies longer than 10 mins result in an error.
  - Click “Upload Video”.
  - Check that progress is displayed in real time.

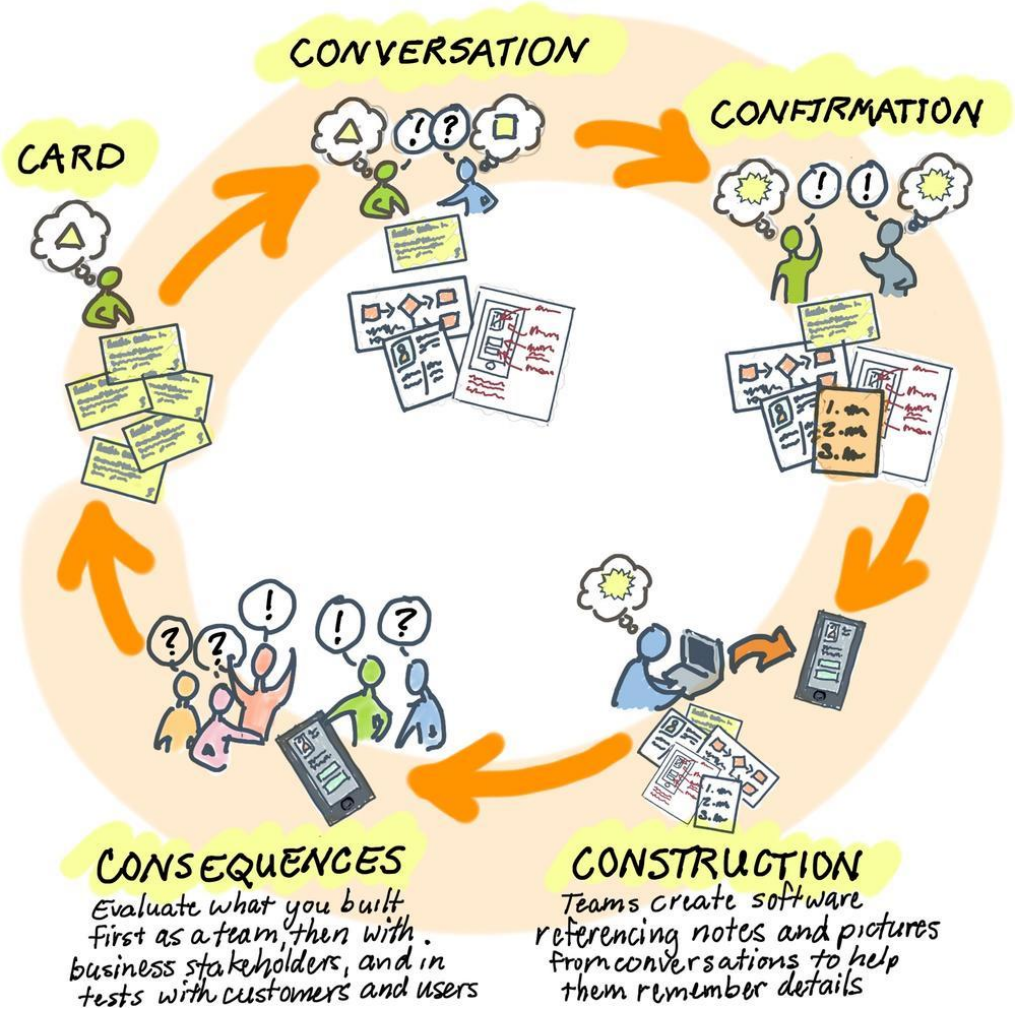
Definition of “Done”

Title:	Customer Inter Account Transfer
Value Statement:	As a bank customer, I want to transfer funds between my linked accounts, So that I can fund my credit card.
Acceptance Criteria:	*Acceptance Criterion 1:* Given that the account is has sufficient funds When the customer requests an inter account transfer Then ensure the source account is debited AND the target account is credited. *Acceptance Criterion 2:* Given that the account is overdrawn, When the customer requests an inter account transfer Then ensure the rejection message is displayed And ensure the money is not transferred.
Definition of Done:	• Unit Tests Passed • Acceptance Criteria Met • Code Reviewed • Functional Tests Passed • Non-Functional Requirements Met • Product Owner Accepts User Story
Owner:	MR I Owner
Iteration:	Unscheduled
Estimate:	5 Points

Different people have different perspectives:

- A programmer might believe “done” is when the code is complete
- A tester is “done” when unit and integration testing is done
- A customer might regard “done” as
  - ✓ Installation, deployment of the system
  - ✓ Together with user documentation
  - ✓ And (perhaps) user training

The Life Cycle of a User Story



User Role Modeling

- On many projects, stories are written as though there is only one type of user. This simplification is a fallacy and can lead a team to miss stories for users who do not fit the general mold of the system's primary user type.
- The following steps can be used to identify and select a useful set of user roles:
  - Brainstorm an initial set of user roles
  - Organize the initial set
  - Consolidate roles
  - Refine the roles

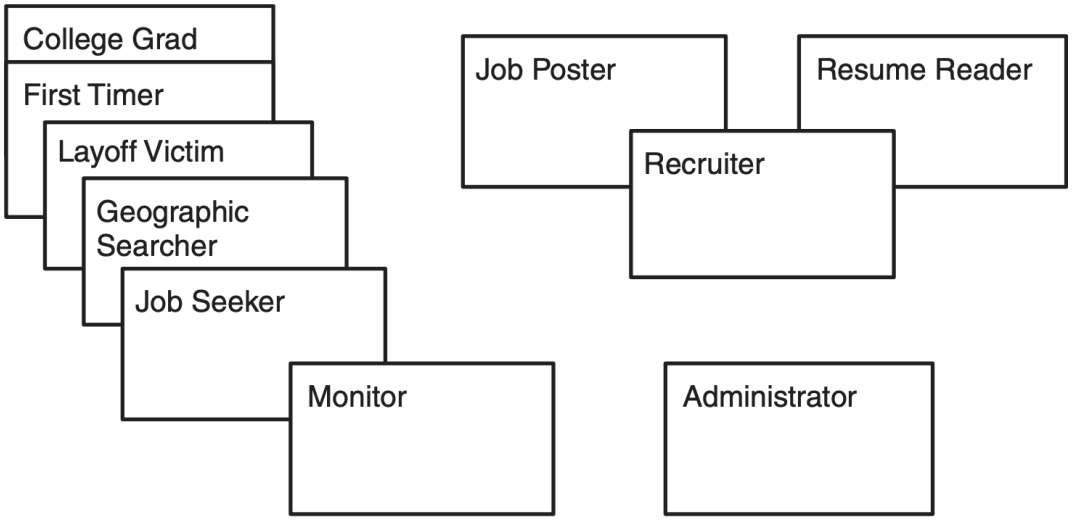
Brainstorming an Initial Set of User Roles

- To identify user roles, the customer and as many of the developers as possible meet in a room with either a large table or a wall to which they can tape or pin cards.
- Each person writes as many cards as he or she can think of. Continue until progress stalls and participants are having a hard time thinking up new roles.

> Suppose we are building the BigMoneyJobs job posting and search site. This type of site will have many different types of users. When we talk about user stories, who is the user we're talking about?

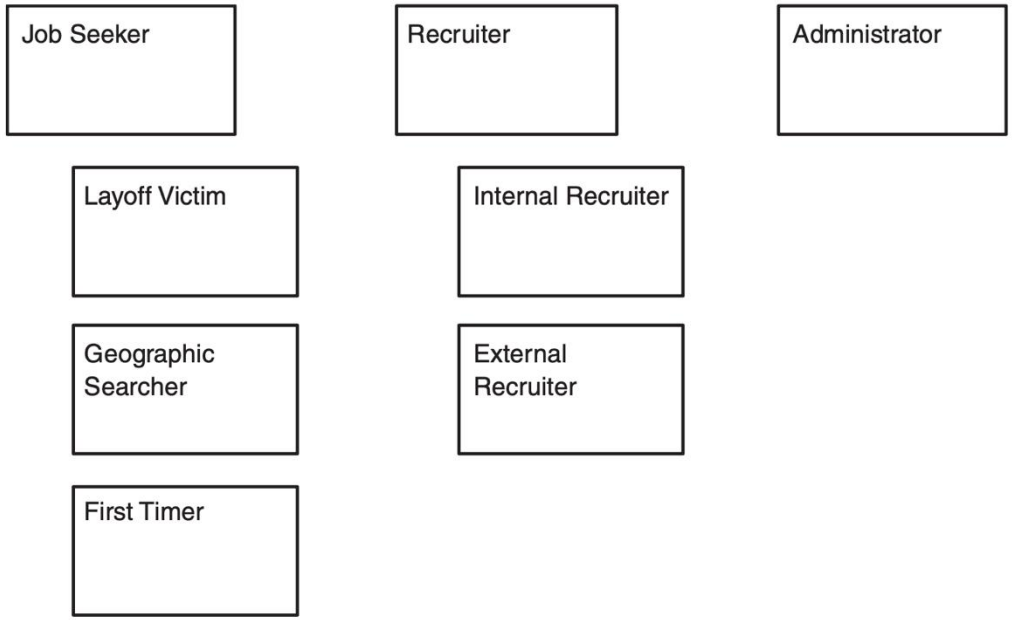
Organizing the Initial Set

- Once the group has finished identify roles, it's time to organize them. To do this, cards are moved around on the table or wall so that their positions indicate the relationships between the roles.
- Overlapping roles are placed so that their cards overlap.
  - If the roles overlap a little, overlap the cards a little.
  - If the roles overlap entirely, overlap the cards entirely.



Consolidating Roles

- ✧ After the roles have been grouped, try to consolidate and condense the roles.
  - If the roles are equivalent, the roles can either be consolidated into a single role, or one of the initial role cards can be ripped up.
  - The group should also rip up any role cards for roles that are unimportant to the success of the system.



Refining the Roles

- Once we've consolidated roles and have a basic understanding for how the roles relate to each other, it is possible to model those roles by defining attributes of each role.
- A role attribute is a fact or bit of useful information about the users who fulfill the role. Any information about the user roles that distinguishes one role from another may be used as a role attribute.

Refining the Roles

✧ Here are some attributes worth considering:

- The frequency with which the user will use the software.
- The user's level of expertise with the domain.
- The user's general level of proficiency with computers and software.
- The user's level of proficiency with the software being developed.
- The user's general goal for using the software. Some users are after convenience, others favor a rich experience, and so on.

User Role: Internal Recruiter

Not particularly computer-savvy but quite adept at using the Web. Will use the software infrequently but intensely. Will read ads from other companies to figure out how to best word her ads. Ease of use is important, but more importantly what she learns must be easily recalled months later.

INVEST

✧ To create good stories we focus on six attributes:

- **I**ndependent
- **N**egotiable
- **V**aluable to users or customers
- **E**stimatable
- **S**mall
- **T**estable

判断用户故事违背哪一个特点

Independent

- Dependencies between stories lead to prioritization and planning problems.
- Dependencies between stories can also make estimation much harder than it needs to be.
- When presented with this type of dependency, there are two ways around it:
  - Combine the dependent stories into one larger but independent story
  - Find a different way of splitting the stories

Negotiable

- Story cards are short descriptions of functionality, the details of which are to be negotiated in a conversation between the customer and the development team.

A company can pay for a job posting with a credit card.

Note: Accept Visa, MasterCard, and American Express. Consider Discover.

VS

A company can pay for a job posting with a credit card.

Note: Accept Visa, MasterCard, and American Express. Consider Discover. On purchases over $100, ask for card ID number from back of card. The system can tell what type of card it is from the first two digits of the card number. The system can store a card number for future use. Collect the expiration month and date of the card.

Valuable to Purchasers or Users

- Keeping in mind the distinction between user (someone who uses the software) and purchaser (someone who purchases the software).
- What you want to avoid are stories that are only valued by developers.

  - All configuration information is read from a central location.
  - All connections to the database are through a connection pool.

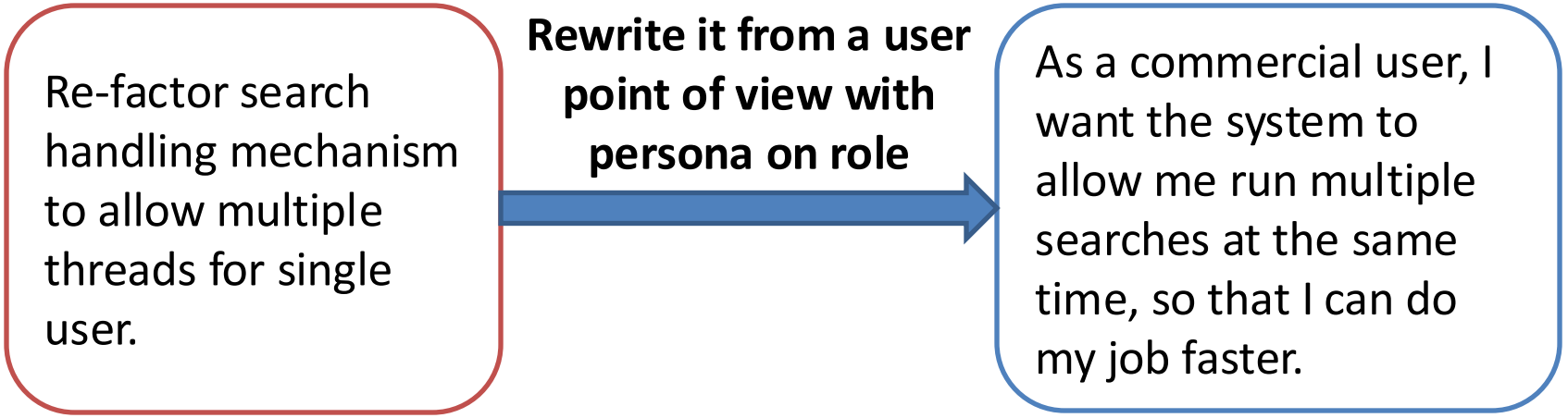
- The best way to ensure that each story is valuable to the customer or users is to have the customer write the stories.

User Story for Users

- ✧ Example: "As a user I want to be able to manage ads, so that I can remove expired and erroneous ads."
- ✧ Who is the user?
  - Is that a portal administrator, who wants to have possibly of database clean-up and ads moderation?
  - Or maybe it is an advertiser, who wants to have list of all ads he submitted to the side and have possibility of removing ads when they are not needed any longer or there was error found in the content?
- ✧ In this case what is missing is persona or role mentioned.

User Story for POs and Developers

- ✧ As a Product Owner, I want the system to have possibility of deleting ads, so that users have possibility of deleting ads.
- ✧ As a developer, I want a database with all the tables to model the data, so I can store information the application needs.



Estimatable

- It is important for developers to be able to estimate (or at least take a guess at) the size of a story or the amount of time it will take to turn a story into working code.
- There are three common reasons why a story may not be estimatable:
  - Developers lack domain knowledge.
  - Developers lack technical knowledge.
  - The story is too big.

A Job Seeker can find a job.

It is sometimes useful to write epics because they serve as place-holders or reminders about big parts of a system that need to be discussed.

Small

- ✧ Story size does matter because if stories are too large or too small you cannot use them in planning.
- ✧ The ultimate determination of whether a story is appropriately sized is based on the team, its capabilities, and the technologies in use.

Splitting Stories

- When a story is too large it is sometimes referred to as an epic.
- Epics can be split into two or more stories of smaller size.

A user can post her resume.

- A user can create resumes, which include education, prior jobs, salary history, publications, presentations, community service, and an objective.
- A user can edit a resume.
- A user can delete a resume.
- A user can have multiple resumes.
- A user can activate and inactivate resumes.

An alternative is to disaggregate along the boundaries of the data.

Combining Stories

- A story that is too small is typically one that the developer says she doesn't want to write down or estimate because doing that may take longer than making the change.
- Bug reports and user interface changes are common examples of stories that are often too small. A good approach for tiny stories, common among Extreme Programming teams, is to combine them into larger stories that represent from about a half-day to several days of work. The combined story is given a name and is then scheduled and worked on just like any other story.

How Detailed Should a User Story Be?

✧ Too broad

- A team member can view iteration status.

✧ Too detailed

- A team member can click a red button to expand the table to include detail, which lists all the tasks, with rank, name, estimate, owner, status.

✧ Just right

- A team member can view the iteration's stories and their status, with main fields.
- A team member can view the current burndown chart on the status page, and can click it for a larger view.
- A team member can view or hide the tasks under the stories.

Testable

- Stories must be written so as to be testable. Successfully passing its tests proves that a story has been successfully developed.
- Whenever possible, tests should be automated. You can almost always automate more than you think you can.
- Examples:
  - A user must find the software easy to use.
  - A user must never have to wait long for any screen to appear.

Guidelines for Good Stories

- ✧ Start with Goal Stories
  - Consider each user role and identify the goals that user has for interacting with our software.
- ✧ Write Closed Stories
  - A closed story is one that finishes with the achievement of a meaningful goal and that allows the user to feel she has accomplished something.

A recruiter can manage the ads she has placed.

a) A recruiter can review resumes from applicants to one of her ads.
b) A recruiter can change the expiration date of an ad.
c) A recruiter can delete an application that is not a good match for a job.

Guidelines for Good Stories

✧ Slice the Cake

A Job Seeker can post a resume

- A Job Seeker can fill out a resume form.
- Information on a resume form is written to the database.

VS

A Job Seeker can post a resume

- A Job Seeker can submit a resume that includes only basic information such as name, address, education history.
- A Job Seeker can submit a resume that includes all information an employer may want to see.

✧ Put Constraints on Cards

- Annotate a story card with “Constraint” for any story that must be obeyed rather than directly implemented.

Guidelines for Good Stories

- ✧ Size the Story to the Horizon
  - Writing stories at different levels based on the implementation horizon of the stories. Stories for the next few iterations would be written at sizes that can be planned into those iterations, while more distant stories could be much larger and less precise.
- ✧ Keep the UI Out as Long as Possible
  - One of the problems that has plagued every approach to software requirements has been mixing requirements with solution specification.
- ✧ Some Things Aren't Stories
  - If you find that some aspect of a system could benefit from expression in a different format, then use that format.

Guidelines for Good Stories

- ✧ Include User Roles in the Stories
  - If the project team has identified user roles, they should make use of them in writing the stories.
- ✧ Write in Active Voice
  - User stories are easier to read and understand when written in active voice.
- ✧ Don’t Number Story Cards
  - Numbering story cards adds pointless overhead to the process and leads us into abstract discussions about features that need to be tangible.

Planning Releases and Iterations

- A project's initial stories are often written in a story writing workshop, but stories can be written at any time throughout the project.
- Collaboratively, the customer team and developers select an iteration length, from perhaps one to four weeks.
- Once an iteration length has been selected, the developers will estimate how much work they'll be able to do per iteration. We call this velocity.
- To plan a release, we sort stories into various piles with each pile representing an iteration. Each pile will contain some number of stories, the estimates for which add up to no more than the estimated velocity.

如何做题

Sample Stories and Their Costs

Story	Story Points
Story A	3
Story B	5
Story C	5
Story D	3
Story E	1
Story F	8
Story G	5
Story H	5
Story I	5
Story J	2

- Suppose that the Table lists all the stories in your project and they are sorted in order of descending priority.
- The team estimates a velocity of thirteen story points per iteration.
- What is the release plan?

Why Change?

- Why not just continue to write requirements documents or use cases?
  - User stories emphasize verbal communication.
  - User stories are comprehensible by everyone.
  - User stories are the right size for planning.
  - User stories work for iterative development.
  - User stories encourage deferring detail.
  - User stories support opportunistic design.
  - User stories encourage participatory design.
  - User stories build up tacit knowledge.

User Story Examples

✧ Explain why the following user stories are not good.

- A user can quickly master the system.
- The software will be released by June 30.
- The system will use Log4J to log all error messages to a file.