# D3 UNIT 1-C
# INTRODUCTION TO PROJECT MANAGEMENT

Product backlog

# User stories
## CCC (*card, conversation and confirmation*)



A user story consists of three parts:

- a written description of the story
- conversations about the story that serve to increase the details of the story
- tests that convey and document details and that can be used to determine when a story is complete

# User stories
## Developers and customers



Developer responsibilities:

- Helping the customer write stories.
- Describing the need in terms of its value to users or the customer.

# User stories
## Developers and customers



Customer responsibilities:

- Writing stories that are promises to converse rather than detailed specifications, have value to users or to yourself, are independent, are testable, and are appropriately sized.

# User stories
## *Customer team*



- Includes those who ensure that the software will meet the needs of its intended users (testers, a product manager, real users, interaction designers…)
- Writes the story cards because they are in the best position to express the desired features and because they must later be able to work out story details with the developers and to prioritize the stories.

# User stories
## Acceptance tests



- Acceptance tests validate that a story has been developed with the functionality the customer team had in mind when they wrote the story.

# User stories
## Acceptance tests



***Example: In an online shopping app…***

"A user can pay for the items in her shopping cart with a credit card.“

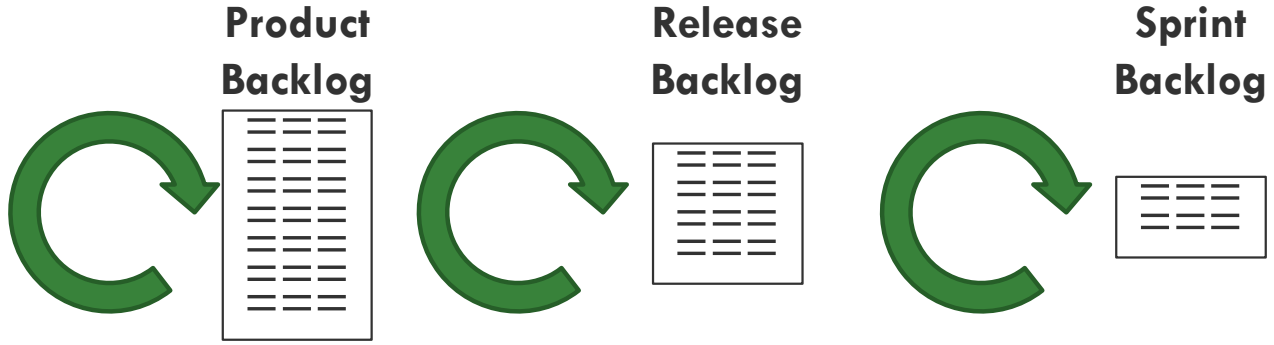
Tests:

- Test with Visa, MasterCard and American Express (pass).
- Test with Diner’s Club (fail).
- Test with a Visa debit card (pass).
- Test with good, bad and missing card ID numbers from the back of the card.
- Test with expired cards.
- Test with different purchase amounts (including one over the card’s limit).

# User stories
## Life cycle



Business goals and user needs




| | | |
|---|---|---|
| As a ___,<br>I want to be able to ___<br>so that ___ | As a ___,<br>I want to be able to ___<br>so that ___ | As a ___,<br>I want to be able to ___<br>so that ___ |
| | I will know this is done when ___ | I will know this is done when ___ |
| | | To do this I must:<br>1. ___<br>2. ___ |

Might have an initial estimate and an expression of technical and business confidence that this is achievable

More detailed estimate, and specific acceptance tests

Development team breaks out the detail of work needed to pass test

Possible automation of the acceptance test

# User stories
## Advantages



- Force a shift to **verbal communication**.
- Provides **rapid feedback** cycles.
- Are **comprehensible** by both developers and users.
- Are the **right size** for planning.
- Work well with **iterative** development.
- Encourage **deferring details**.
- Encourage **opportunistic development**, in which the team readily shifts focus between high and low levels of detail as opportunities are discovered.
- Encourage **participatory design**, in which users become active and valued participants.

# User stories
## Disadvantages



- **On large projects** it can be **difficult to keep** hundreds or thousands of stories organized; they may need to be augmented with additional documents for traceability.
- **Conversations do not** scale adequately to **entirely replace written documents** on large projects.

# User stories



A. The user can run the system on Windows 10/11 and Linux.
B. All graphing and charting will be done using a third-party library.
C. The user can undo up to fifty commands.
D. The software will be released by June 30.
E. The software will be written in Java.
F. The user can select her country from a drop-down list.
G. The system will use Log4J to log all error messages to a file.
H. The user will be prompted to save their work if they haven't saved it for 15 minutes.
I. The user can export data to XML.

# Writing user stories
## INVEST (Bill Wake)



- Independent
- Negotiable
- Valuable to users or customers
- Estimatable
- Small
- Testable

# Writing user stories
## Independent



Ideally, stories are independent (this is not always possible)

1. A company can pay for a job posting with a Visa card.
2. A company can pay for a job posting with a MasterCard.
3. A company can pay for a job posting with an American Express card.

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

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

# Writing user stories

## Negotiable



Details are to be negotiated in a conversation between the customer and the development team. Details that have already been determined through conversations become tests. Tests can be noted on the back of the story card.

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.

# Writing user stories
## Negotiable



Details are to be negotiated in a conversation between the customer and the development team. Details that have already been determined through conversations become tests. Tests can be noted on the back of the story card.

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

Note: Will we accept Discover cards?
Note for UI: Don't have a field for card type (it can be derived from first two digits on the card).

Test with Visa, MasterCard and American Express (pass).
Test with Diner's Club (fail).
Test with good, bad and missing card ID numbers.
Test with expired cards.
Test with over $100 and under $100.

# Writing user stories
## Valuable to purchasers or users



Many projects include stories that are not valued by users:

- All configuration information is read from a central location.
- Throughout the development process, the development team will produce documentation suitable for an ISO 9001 audit.
- The development team will produce the software in accordance with CMM Level 3 *(Capability Maturity Model).*

# Writing user stories
## Valuable to purchasers or users



- All connections to the database are through a connection pool.
- All error handling and logging is done through a set of common classes.

- Up to fifty users should be able to use the application with a five-user database license.
- All errors are presented to the user and logged in a consistent manner.

# Writing user stories
## Estimatable



Reasons why a story can not be estimatable:

- Developers lack domain knowledge.
- Developers lack technical knowledge.
- The story is too big (epic).

# Writing user stories
## Estimatable



- Functional spikes (what?) – They are used to analyze overall solution behavior and determine:
  - How to break it down
  - How to organize the work
  - Where risk and complexity exist
  - How to use insights to influence implementation decisions
- Technical spikes (how?)– They are used to research various approaches in the solution domain. For example:
  - Determine a build-versus-buy decision
  - Evaluate the potential performance or load impact of a new user story
  - Evaluate specific technical implementation approaches
  - Develop confidence about the desired solution path

# Writing user stories
## Estimatable



"As a consumer, I want to see my daily energy use in a histogram so that I can quickly understand my past, current, and projected energy consumption”

Functional spike – Prototype a histogram in the web portal and get some user feedback on presentation size, style, and charting

Technical spike - Research how long it takes to update a customer display to current usage, determining communication requirements, bandwidth, and whether to push or pull the data

# Writing user stories
## Estimatable



“Add novel extensions to standard expectation maximization”

- spike story (“research and determine the feasibility of extending expectation maximization”)
- functional story (“extend expectation maximization”)

# Writing user stories
## Small



If stories are too small, they can be combined into a larger one.

If they are too large (epics), they typically fall into one of two categories:

- The compound story
- The complex story

# Writing user stories
## Small



***Example: In a job search site…***

"A user can post their resume."

***Disaggregation 1 (by operation: create, edit, delete)***

- 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.

# Writing user stories
## Small



***Example: In a job search site…***

"A user can post their resume"

***Disaggregation 2 (by data)***

- A user can add and edit education information.
- A user can add and edit job history information.
- A user can add and edit salary history information.
- A user can add and edit publications.
- A user can add and edit presentations.
- A user can add and edit community service.
- A user can add and edit an objective.

# Writing user stories
## Small



“A user can search for jobs”

Details:

- What values can users search on? State? City? Job title? Keywords?
- Does the user have to be a member of the site?
- Can search parameters be saved?
- What information is displayed for matching jobs?

# Writing user stories
## Small



"A user can search for jobs"

- A user can search for jobs by attributes like location, salary range, job title, company name, and the date the job was posted.
- A user can view information about each job that is matched by a search.
- A user can view detailed information about a company that has posted a job.

A user can view a job description.
A user can view a job's salary range.
A user can view the location of a job.

# Writing user stories

## Small



Users can view information about each job that is matched by a search.

Marco says show description, salary, and location.

Try it with an empty job description.
Try it with a really long job description.
Try it with a missing salary.
Try it with a six-digit salary.

# Writing user stories
## Testable



A user must find the software easy to use.

A user must never have to wait long for any screen to appear.

"A novice user is able to complete common workflows without training “

(can be tested, but not automatically)

# Writing user stories



A. A user can quickly master the system.
B. A user can edit the address on a resume.
C. A user can add, edit and delete multiple resumes.
D. The system can calculate saddlepoint approximations for distributions of quadratic forms in normal variables.
E. All runtime errors are logged in a consistent manner.

# Writing user stories



***Example: In a job search site…***

"A user can make and change automated job search agents. "
*(epic)*

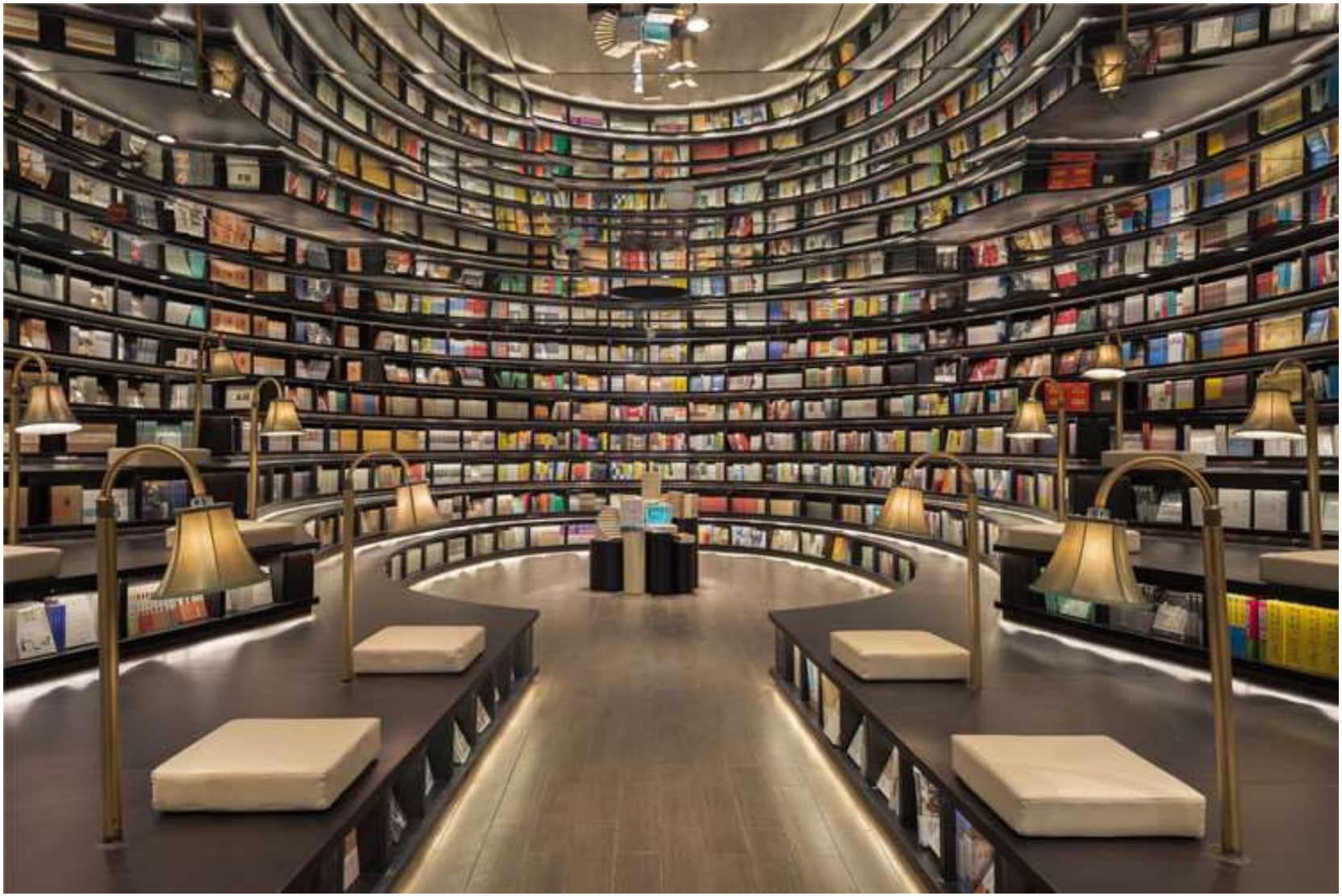
- A user can make an automated job search agent.
- A user can edit the search parameters of an automated job search agent.

optional

- A user can change the times when an automated job search agent will run.
- A user can change how the results of an automated job search agent will be reported.

# Example
## Bookshop





# Example
## User stories



- *A user can search for books by author, title or ISBN number.*
- *A user can view detailed information on a book. For example, number of pages, publication date and a brief description.*
- *A user can put books into a "shopping cart" and buy them when they are done shopping.*
- *A user can remove books from their cart before completing an order.*
- *To buy a book the user enters their billing address, the shipping address and credit card information.*

A user can do a basic simple search that searches for a word or phrase in both the author and title fields.

A user can search for books by entering values in any combination of author, title and ISBN.

# Example
## User stories



- *A user can rate and review books.*
- *A user can establish an account that remembers shipping and billing information.*
- *A user can edit their account information (credit card, shipping address, billing address and so on).*
- *A user can put books into a "wish list" that is visible to other site visitors..*

| |
|---|
| A user can rate books from 1 (bad) to 5 (good). The book does not have to be one the user bought from us. |
| A user can write a review of a book. They can preview the review before submitting it. The book does not have to be one the user bought from us. |

| |
|---|
| A user can edit the credit card information stored in their account. |
| A user can edit the shipping and billing addresses stored in their account. |

# Example
## User stories



- *A user can place an item from a wish list (even someone else's) into their shopping cart.*
- *A repeat customer must be able to find one book and complete an order in less than 90 seconds.* constraint
- *A user can view a history of all of their past orders.*
- *A user can easily re-purchase items when viewing past orders.*
- *The site always tells a shopper what the last 3 (?) items they viewed are and provides links back to them (this works even between sessions).*
- *A user can see what books we recommend on a variety of topics.*

# Example
## User stories



- *A user can easily find the wish lists of other users.*
- *A user can choose to have items gift wrapped.*
- *A user can choose to enclose a gift card and can write their own message for the card.*
- A Report Viewer can see reports of daily purchases broken down by book category, traffic, best- and worst-selling books and so on.
- A user must be properly authenticated before viewing reports.
- Orders made on the website have to end up in the same order database as telephone orders. constraint

A user can search for a wish list based on its owner's name and state.

# Example
## User stories



- An administrator can add new books to the site.
- An administrator needs to approve or reject reviews before they are available on the site.
- An administrator can delete a book.
- An administrator can edit the information about an existing book.
- A user can check the status of their recent orders. If an order has not shipped, they can add or remove books, change the shipping method, the delivery address and the credit card.
- The system must support peak usage of up to 50 concurrent users. constraint

| |
|---|
| A user can check the status of their recent orders. |
| If an order has not shipped, a user can add or remove books, change the shipping method, the delivery address and the credit card. |

# References



"User Stories Applied For agile software development"

Mike Crohn

The Addison Wesley signature series

2009