Tasking a Story

MARTIN LAPOINTE • SENIOR AGILE COACH •
YELLOW PAGES CANADA • @MDELAPOINTE

What size should a user story be?

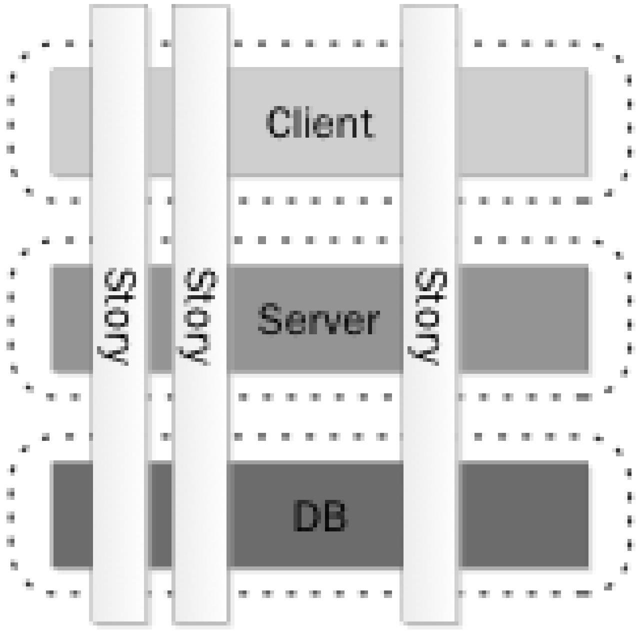
- A story should be small enough to be coded and tested within an iteration—ideally just a few days.
- When a story is too large, it is called an epic. Backlog items tend to start as epics.

Why split a Story into tasks ?

1. The team finds out all WHAT needs to be done for each user stories.
2. Multiple developers can work on the same story in Scrum.
3. It allows us to measure progress.

A piece of working software

To accomplish this, you will have to pass through the UI layer, service layer, persist some data in the data layer, and make an API calls, etc..



Example

In order to easily be able to follow my friends' tweets, as a registered user, I want to automatically follow all of my gmail contacts that have twitter accounts.

Your tasks might be:

- Add an option to the menu
- Add a new gmail authentication screen
- Add a twitter authentication screen
- Add a contact selection screen
- Add a controller that calls into your service layer
- Save contacts to the database
- Modify your existing gmail API calling service to get contacts
- Add a twitter API calling service to follow selected contacts

Working with acceptance criteria

Allows us to:

- Think "from the outside in"
  - In other words implement only those behaviors which contribute most directly to the expected business outcomes, so as to minimize waste.

- Describe behaviors in a single notation which is directly accessible to domain experts, testers and developers to improve communication.

- Gather in a single place the specification of an outcome valuable to a user, as well as examples or scenarios expressed.

How to write better tasks

Write "functional" task

Imagine an objective "the story” and then write an algorithm in order to fulfil the objective.

Impacts :

- You are sure to not forget anything, and to don't do too much
- Possible to measure progress
- Tasks really say what to do and should be easily understandable by any team member, or even by people outside the team
- Tasks are independent and the team can work together on the same story
- Tasks are testable and so issues and bugs are noticed earlier

“Given - When - Then” model

The Given-When-Then formula is a template intended to guide the writing of acceptance criteria for a User Story:

- (Given) some context
- (When) some action is carried out
- (Then) a particular set of observable consequences should obtain

An example:

- When I attempt to withdraw an amount less than my card's limit, Then the withdrawal should complete without errors or warnings.

Example

Let’s use the classic example of an ATM machine:

Customer withdraws cash

As a customer, *I want* to withdraw cash from an ATM, *so that* I don’t have to wait in line at the bank.

- There are several scenarios to consider
- Let’s take one for the example

Using the given-when-then template

Scenario 1: Account is in credit

- *Given* the account is in credit
- *And* the card is valid
- *And* the dispenser contains cash
- *When* the customer requests cash
- *Then* ensure the account is debited
- *And* ensure cash is dispensed
- *And* ensure the card is returned

Using the given-when-then template

Acceptance criteria (behaviour oriented):

- Verify that the card is valid
- Validate the account is in credit
- Allow the customer to requests cash
- Ensure that the account is debited
- Dispense cash based on the amount debited
- Upon completion of transaction, return card to customer

Using the given-when-then template

Tasks from the team (behaviour oriented):

- Read card and validate serials to authenticate
- Connect to system x to validate account is in credit
- Send request to system x with $ parameters to requests cash
- Get confirmation code from system x to confirm account is debited
- Trigger function y to Dispense cash
- Trigger function z to return card to customer

Conclusion

Tasks :

- Are the response of expected behaviours
- Are in the form of working and testable units
- Should be fine grained (7h-14h max)
- Are independent
- Allow multiple team members to work simultaneously on a User Story
- Used to validate the scope of a Story