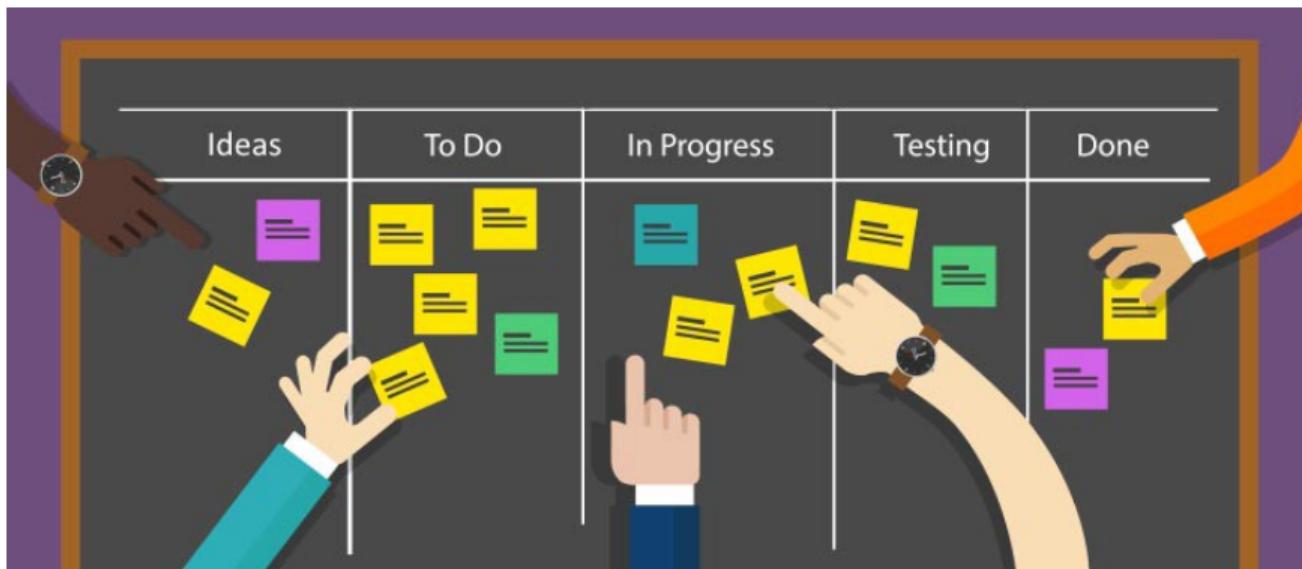


Another way of writing requirements

Agile Development: User stories are the new requirements document



User Story

- Requirements can also be written as User Stories



A definition given by Martin Fowler

User stories are a chunk of functionality that is of value to the customer.

Functionality, it's the key word here.

User stories should be written using business language. They must be functional and state clearly what it is expected, not necessarily in detail but in purpose.

Guidelines

1. Each User story should state one thing well

Example

As an applicant, I can navigate to the coverage screen, enter personal and vehicle data, and submit the application online to request car insurance cover.



Clearer

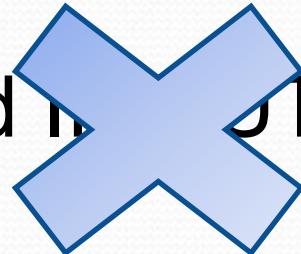


As an applicant, I can navigate to the coverage screen to select the insurance coverage I need.

As an applicant, I can enter personal and vehicle data to compare premiums

As an applicant, I can submit an application online to request car insurance cover.

Avoid OR



Only use AND to connect data

Guidelines

1. Keep it Simple – the user story
 - State one thing, well
2. What not How

What Not How

Limiting the choices

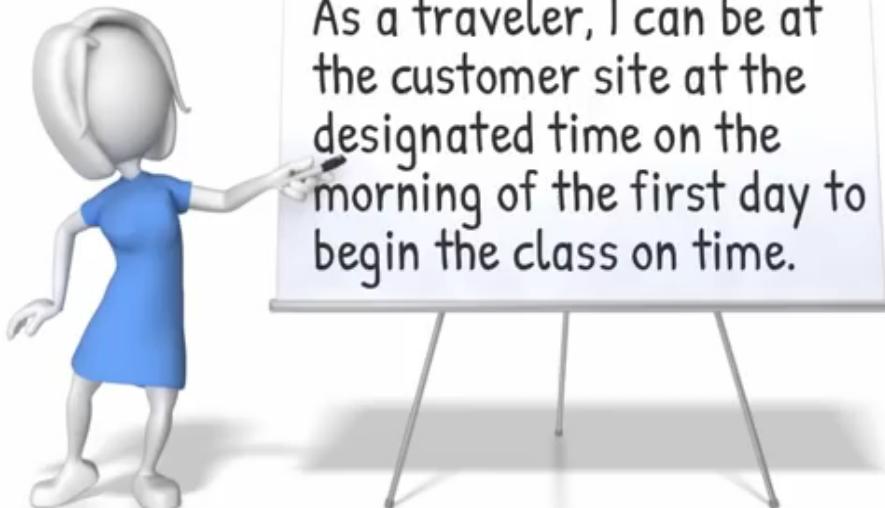


How / Solution



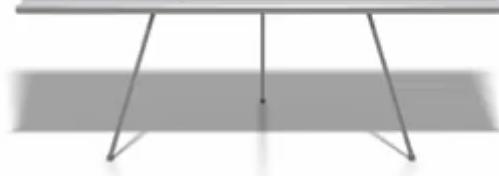
Booking a flight

Better to leave lots of o



Example

As an applicant I can select my country from a drop down box to avoid entering an invalid country



Better

As an applicant I can select a valid country to ensure an accurate quote for insurance cover



Rule 2

An effective user story
expresses WHAT should
be done, not HOW
to accomplish it.



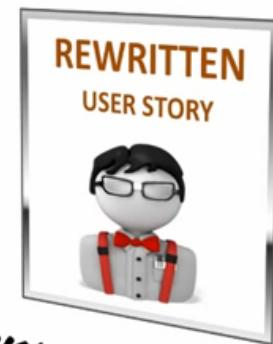
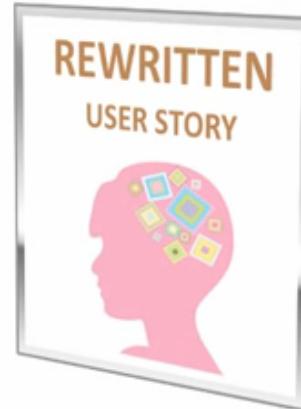
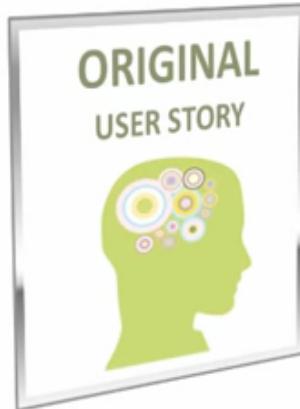
Guidelines

1. Keep it Simple – the user story
 - State one thing, well
2. **What not How**
3. Make sure the user story is relevant
 - Within the scope of the project



Guidelines

1. Keep it Simple – the user story
 - State one thing, well
2. What not How
3. Make sure the user story is relevant
 - Within the scope of the project
4. Avoid ambiguity



DIFFERENT THINKING STYLES

Avoid Ambiguity



Guidelines

1. Keep it Simple – the user story
 - State one thing, well
2. What not How
3. Make sure the user story is relevant
 - Within the scope of the project
4. Avoid ambiguity
5. Add non functional requirements to add value to the user stories

User Story

- Requirements can also be written as User Stories



As a Role the author represents,

I / We can do or have something

With these qualities

To achieve my goal or objective

Functional

Non Functional

This is EPIC

- An Epic is a large user story and MUST be broken down before it can be developed.