
Story Writing

Software Engineering 2 – Workshop #1

Agenda

- Introduction to User stories and
 - Presentation of the case study (10 min)
 - Team work (1 h)
 - Wrap-up (10 min)
-

USER STORIES

User Story

- A description of an interaction between a user and the system to be developed
 - A promise for a conversation
 - Must be:
 - Testable
 - Valuable
-

Story

- Simple format:

As a *<actor type>*

I want *<to do something>*

So that *<some value is created>*

Story – example

As a customer

I want to use my Credit Card

So that I can pay for the items

Story – example

As a *user*

I want *to add items*

~~So that~~ *to my wish list*

Stories vs. Epics

- Story
 - Manageable piece of requirement
 - It is INVEST
 - Epic
 - Too large a story
 - Needs to be split into stories
 - Splitting yields smaller stories that still bring value to the users
-

Epic – example

As a customer

I want to be able to view and modify wish-list

So that I can decide later which items to buy

Case Study

OFFICE QUEUE MANAGEMENT SYSTEM

Office Queue Management

- The Office Queue Management is a system that manages the queues for desk services open to the public (e.g., post office, medical office).
 - In the same office, various counters can handle different types of services (e.g., shipping, accounts, mail, etc.).
 - When a client gets to the office, they specify the service type and receive a ticket with a wait code.
 - Ticket codes are unique and will be used to call clients to the correct counter when their request can be served.
-

Groups

- Compose groups of 5–7 members with your neighbors and move a little so that you can sit close to each other
 - Write your stories on post-it
 - You can ask questions we'll be around
 - We will start going around group by group in ~15 min
 - General wrap-up at 11:15
-

Register your presence here



To WORK!

FAQ

- The actor can be the system?
 - No: the actor is always external to the system
 - The client is only one that has a ticker?
 - No: the system also provide info to clients without a ticket, e.g. waiting time
-

Next: @ 11:10

WRAP-UP

Reminders 1 / 2

- The system should never be an actor: stories describes how a **human** actor benefit from the system supporting her in performing a task
 - If you happen to write a story with the system as actor, think of who's going to benefit from the function: that is the actor
 - Remember the story concerns the system
 - If your story does not involve the software product probably it should not end up in the *product* log
 - Remember INVEST: review your stories!
 - check whether you should merge some highly dependent stories
 - move details on the back of the post-it
-

Reminders 2 / 2

- Remember: Actor → action → value
 - Stories have only **one** actor
 - Respect the form “I want to + verb” , do not use “I need” , “I shall”, etc.
 - The story is driven by some value for stakeholders (*so that*)
 - Use active forms of the verbs (as much as possible)
 - As an officer I want to call the next customer so that she can be served
 - As an officer I want to call the next customer so that I can serve her
 - Today you are customer (no tech knowledge) next time you will get back to being software engineers
-

What is a right value

- A typical story could be

As a customer
I want to select a service
So that I get a ticket

- Alien speaking: “What is the value in getting a ticket? Is it like a lottery?”
 - We ought to have a shared knowledge of how a queue with tickets works to understand
-

What is a value

- Values are not bound to the ideal world, but they have very pragmatic effects not only on the software implementation, but on the final users too.
- Example, compare the implications of the values:

As a manager

I want to see the **daily/weekly** service stats

So that I can better control

- the work of the officers
 - the execution of the services
-

Separated or joined

- To decide whether two related or similar stories should be separate or joined is whether they represent
 - values that could be delivered separately or
 - faces of the same feature that should be delivered jointly
-

To be merged

To be merged	As a manager I want to see the monthly service stats	As a manager I want to see the daily service stats
To be kept separate	As a customer I want to select a service So that I can be eventually served	As a customer I want to select a service So that I know the relative estimated waiting time

The stories you will use from next week

Get ticket

As a customer

I want to select a service

So that I get served when my turn comes

Next customer

As an officer

I want to call the next client to my counter

So that I can serve him or her

Call customer

As a customer

I want to know when my turn comes and where to go

See stats

As a manager

I want to see the daily/weekly service stats

So that I can better control the office procedures

Config counters

As an administrator

I want to configure the counter and services

To properly start the daily office work

Get estimated time

As a customer

I want to know the estimated waiting time for a service

Notify customer served

As an officer

I want to signal a customer has been served

So that proper statistics can be gathered
