COMS W4156 Advanced Software Engineering (ASE)

September 28, 2021

shared google doc for discussion during class

Team Project

Team Formation due yesterday

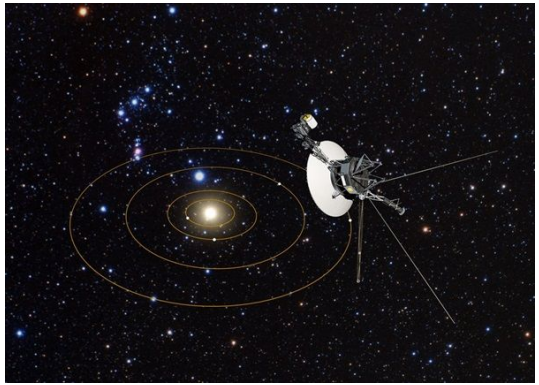
Is there anyone here who does not yet have a team or has a team with less than four members?

Software Requirements

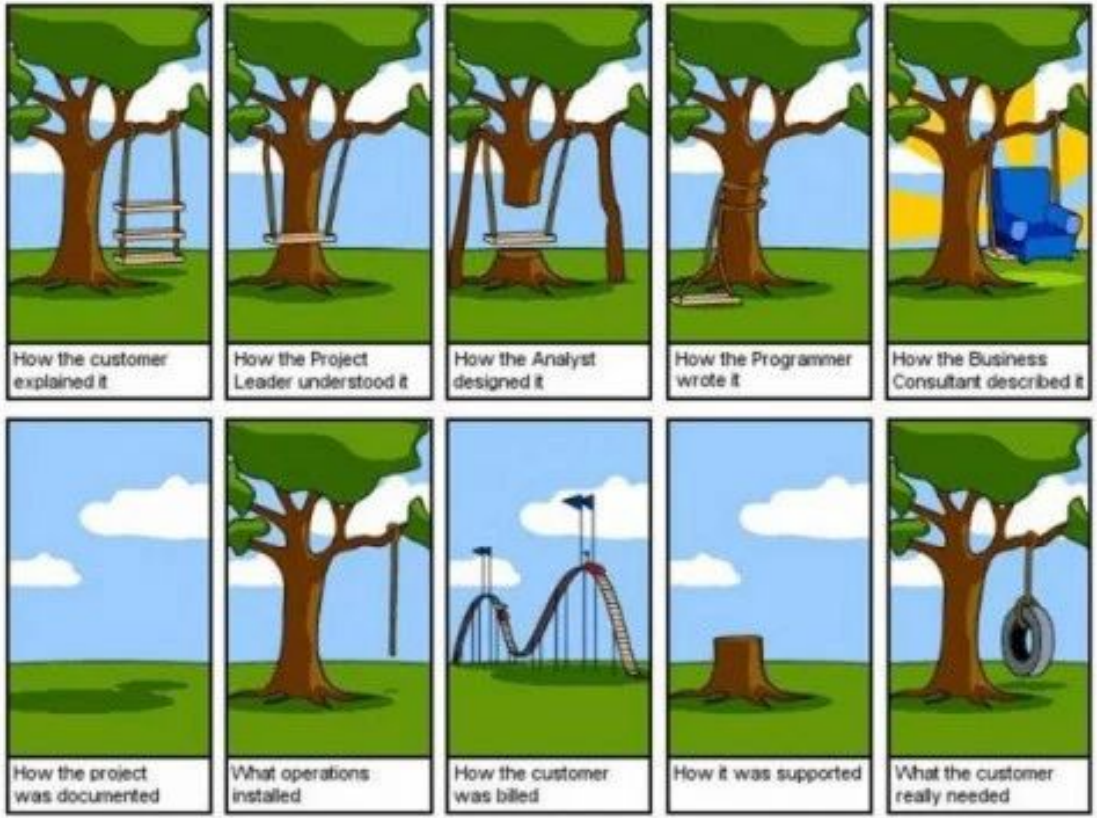
Software Requirements Specifications (SRS) describe requirements in immense detail, e.g., see HTTP, URL

Detailed specifications are necessary for interoperability across multiple vendors, e.g., so internet software, like web browsers and servers, can talk to each other

Detailed specifications are also necessary for software where you have to get it right the first time, e.g., NASA Voyager Telecommunications



But Customers Don’t Always Know What They Want



Customers vs. Users

Customer is not necessarily the same as *user*

If you are developing software to control a self-driving car, your customer might be Waymo or Tesla, but your primary user is the person sitting behind the wheel of the car - that user is your customer's customer

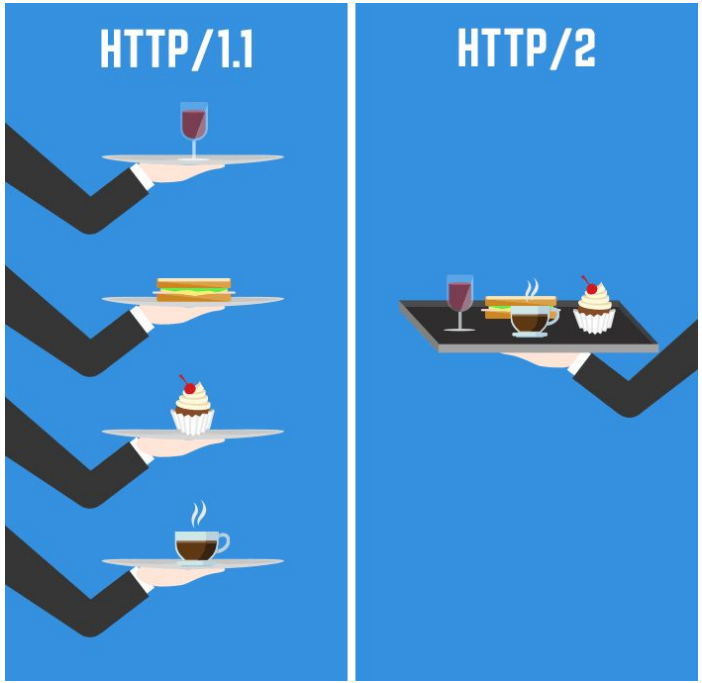
Many software teams have internal "*product owners*" who proxy for customers and users

We'll often use the terms customer and user interchangeably, and not distinguish whether requirements come from the customer/user or from the product owner

Most Software Requirements Change

Even internet software changes:
HTTP/0.9 ⇒ HTTP 1.0 ⇒ HTTP/1.1 ⇒ HTTP/2.0 ⇒ HTTP/3?

Slow evolution of detailed internet standards is enabled by the IETF standards process, but it (intentionally) cannot support rapid change

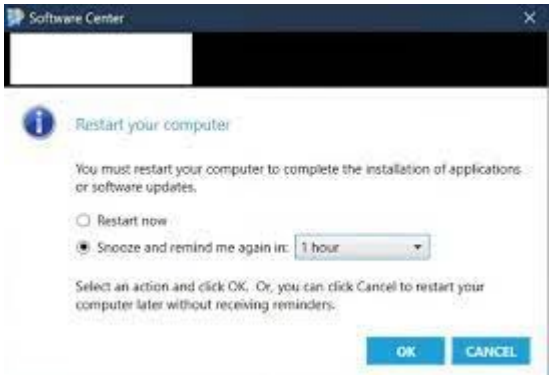


Most Software Requirements Change

Almost all modern business and consumer software delivers frequent updates

Google and Facebook update their user-facing software multiple times **per day**

These organizations are not updating RFC-like documents, they are using shorter, simpler requirements documents and processes

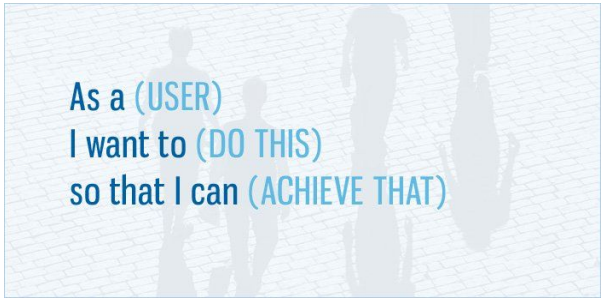


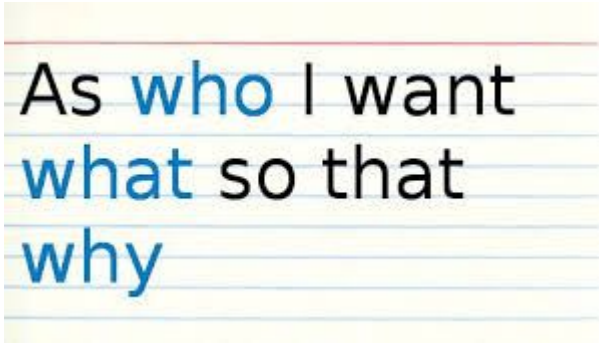
User Story ~= Feature

User stories describe what users will be able to do with the software that they couldn't do without it, or couldn't do as well without it

A user story is a short simple description of ONE feature, ideally fitting on a 3x5 card, written in customer's (or user's) language

< label >: As a < type of user >, I want < some goal > so that < some reason >





User Stories

Conditions of satisfaction (acceptance criteria) might or might not fit on the back of a 3x5 card, but are also written in customer's/user's language

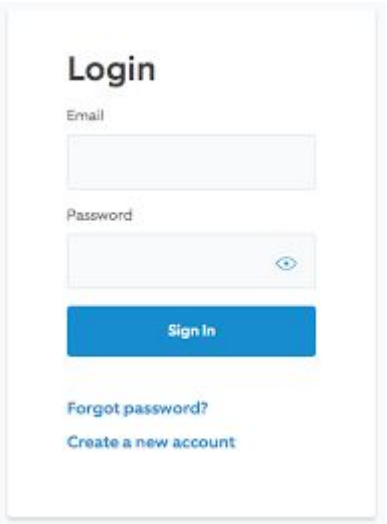
< *label* >: *As a < type of user >, I want < some goal > so that < some reason >*

My conditions of satisfaction are < list of common cases and special cases that must work >

Why type of user? Many applications have multiple *roles* that use the software differently, such as performing different tasks (e.g., instructor vs. student using Courseworks)

Why customer's language? The customer probably did not ask you to use json or gradle, but may require that the new application runs on existing hardware, draws data from existing database, and interfaces to existing applications (e.g., Courseworks gets rosters from ssol)

Does This Story Say Enough to Start Coding?



As a user, I am required to login before using the site

- user is logged in only when proper credentials are provided
- a "remember me" option is available
- user can request a password reminder
- user is locked out after three failed attempts

What other information might the developers need?

How would they estimate how many person-hours are required to develop?

Does This Story Say Enough to Start Coding?

As a vice president of marketing, I want to select a holiday season to be used when reviewing the performance of past advertising campaigns so that I can identify the most profitable ones

- Make sure it works with major retail holidays: Christmas, Easter, President's Day, Mother's Day, Father's Day, Labor Day, New Year's Day.
- Support holidays that span two calendar years (Christmas through New Year's).
- Holiday seasons can be set from one holiday to the next (Thanksgiving to Christmas).
- Holiday seasons can be set to be a number of days prior to the holiday.

What other information might the developers need?

How would they estimate how many person-hours are required to develop?

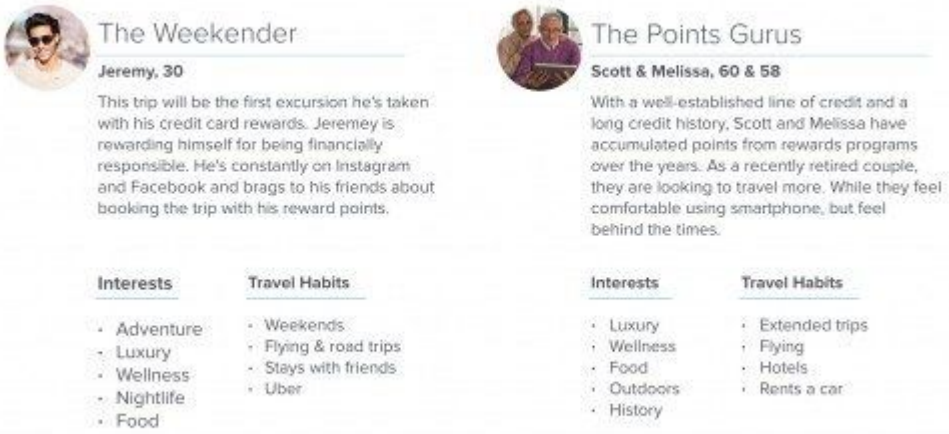
User Personas

Synthetic biographies of fictitious users, usually one for each category likely to have different preferences and expectations of the product (!= user roles)

Personas are an answer to the observation that a designer who tries to please everybody ends up pleasing nobody, because too many compromises kill the product's integrity

"What would Jeremy do?"

"Will Scott & Melissa understand this interaction or how to use this feature?"

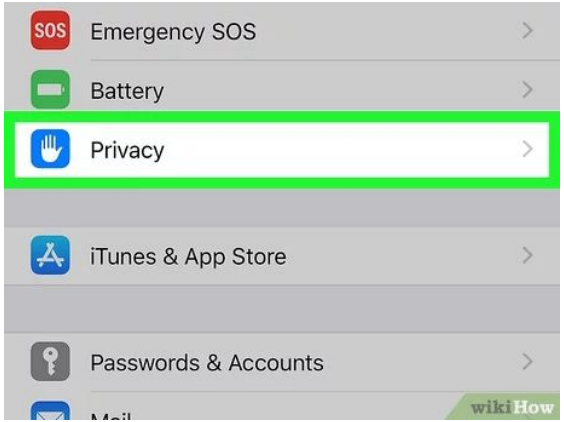


Does This Story Say Enough to Start Coding?

As a traveler, I want to know which nearby shops participate in my rewards program, so I can buy stuff

Conditions of Satisfaction:

- If my location service is not already turned on, it should prompt me to turn it on
- It should allow me to input a location address instead
- “What would Jeremy do?”
- “Will Scott & Melissa understand this interaction or how to use this feature?”



What other information might the developers need?

How would they estimate how many person-hours are required to develop?

User Stories for Services (we'll call them "Service Stories")

Services are built for software engineers, to use in their own applications and services

There's no point in developing a service unless you have some ideas about what kinds of applications (or other services) might use it and what they would use it for - service personas are fictitious examples

Service personas would be presented differently than user personas - e.g., probably not including stock photos and hobbies

While user stories should describe functionality rather than the data needed to implement that functionality, avoiding CRUD (create, read, update, delete), CRUD might be exactly what is needed from some service APIs

Does This Story Say Enough to Start Coding?

As a visitor shopping app, I want an API that returns the list of nearby shops when I send a location, so that I can refer to these shops

Conditions of Satisfaction

- The API should accept latitude, longitude and (optional) max-distance
- It should return a list of shops whose entrances are within max-distance (or default) of the requested lat-lon

What other information might the developers need?

How would they estimate how many person-hours are required to develop?

Does This Story Say Enough to Start Coding?

As a rewards app, I want an API that returns the rewards programs affiliated with a specified shop, so that I can determine whether that shop is relevant

Conditions of Satisfaction

- The API should accept the name and address of a specific shop
- It should return the list of rewards programs the shop participates in

What other information might the developers need?

How would they estimate how many person-hours are required to develop?

In-Class Exercise

Courseworks, aka Canvas, has a GUI - if you read the course home page or read/submitted any assignments, you have used the GUI

Canvas also has a REST API (posted here), which you probably have not used

Your goal is to develop an app (or service) called CONTAIN, for **CON**tact **T**racing for instruction **A**ss**I**sta**N**ts, which does “contact tracing” based on IA (instruction assistant) connections. An IA for xxx course is a contact of all the other IAs for xxx as well as for every student who takes xxx. Since an IA is also a student, they are a contact for all other students in every course they take as well as for all the IAs of those courses

Develop service stories for Canvas to provide all the data you need to implement your CONTAIN app. Canvas already knows who all the IAs are, assuming the instructor entered this information - but ssol does not. Make sure your stories are for features of the Canvas API, not user stories for CONTAIN. Make up anything plausible for Canvas, don’t worry about what it really provides. Ignore for now instructors, CVN/CUIT staff, facilities staff, and anyone else who might enter the classroom that is not a student

Blank paper available

timer..

Individual Mini-Project

Three parts:

1. Implementing a simple game (past due)
2. Testing the game (due tomorrow unless you started the course late)
3. Saving game state (due October 6)

See Connect Four

Team Project

Preliminary Project Proposal due October 13