

Backlog Grooming

ACTIVITY

1. **You need to identify epics, stories and tasks for your given topic. (10 marks)**
 - a. Minimum 6 epics per team
 - b. Minimum 5 stories per epic
 - c. Minimum 6 tasks per story
 - d. Development tasks in each epic by team members handling developer role
 - e. Testing tasks in each epic by team members handling test engineer role
 - f. General Engineering tasks by team
 - g. Identify priority using MoSCoW method
 - h. Enter estimates following the given guidelines
2. **Use chart papers, sticky notes and color pens to showcase the epics/stories (5 marks for individual epics (by subject handling faculty) and 5 marks for overall appeal/team effort by external faculty)**
 - a. One epic per team member on the chart paper
 - b. Epics and stories are sufficient. Tasks need not be written on the chart
 - c. Chart top right corner should mention team name and team member name/roll number
 - d. Prepare the chart as creative as your team can

Time Given: One week from the date of release of this assignment

Writing User Stories – Guidelines, Tips and examples

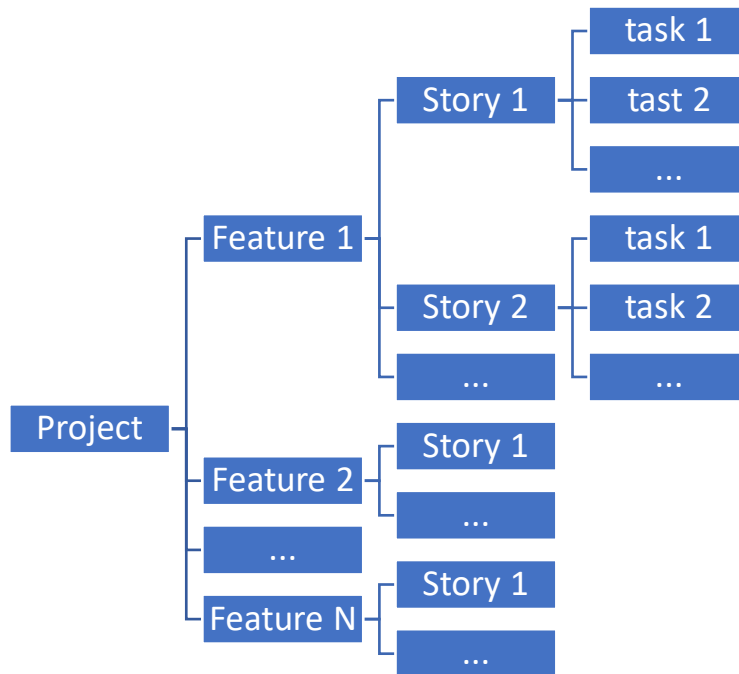
Reference materials

- Lecture notes: 08_09-Agile.pptx.
- <https://www.kbp.media/themes-epics-features-user-stories/>
- <https://luis-goncalves.com/epic-user-story-task/>
- <https://mohamedradwan.com/2017/12/05/requirements-epic-feature-user-story-task-size-and-estimation-in-agile-and-scrum/>

Summary of User stories

- User Stories combine written and verbal communications, supported with a picture where possible.
- User Stories should describe features that are of value to the user, written in the user's language.
- User Stories detail just enough information and no more.

- Details are deferred and captured through collaboration just in time for development.
- Test cases should be written before development, when the User Story is written.
- User Stories should be Independent, Negotiable, Valuable, Estimatable, Small and Testable.



- **Epic/Feature** - A general use case that is a collection of user stories
- **User Story** - A concise, written description of a piece of functionality that will be valuable to a user (or owner) of the software
- **Task** - Represents development tasks to accomplish the user story. Generally no more than 1-day tasks.
- (Engineering) **Task** - represents a set of engineering work that is not directly related to a user story.

EXAMPLE:

- **Epic:** User Authentication.
- **User Stories:**
 - User Login screen.
 - Forgot Password workflow.
 - Lock account after too many failed attempts.
 - Google login support.
 - Facebook login support.
- **Sub-Tasks:**
 - User Login screen:
 - Design login page.
 - Cut SVG icons and images.

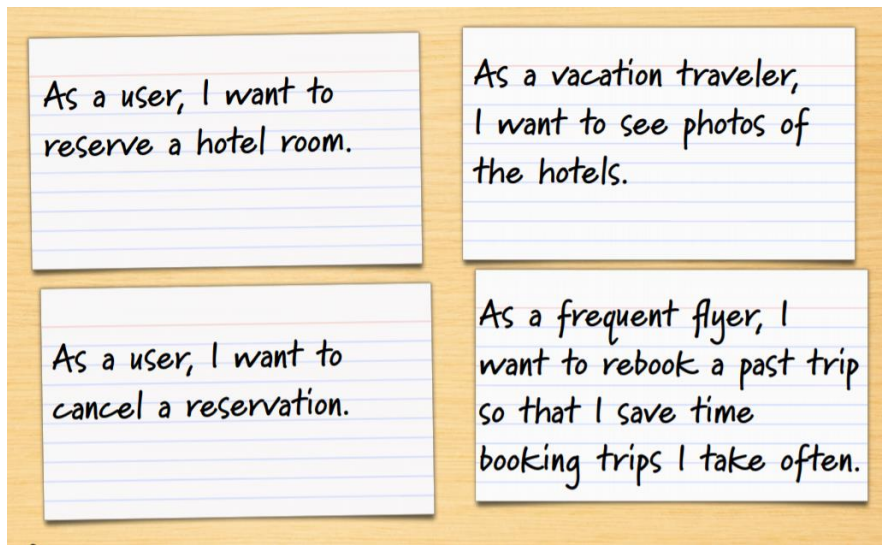
- Implement login page HTML/CSS/JS.
 - Create SQL scripts to create tables.
 - Create SQL scripts for stored procedures.
 - Create web service REST API for user resource.
 - Hook up login page to web service REST API.
- Forgot Password workflow:
 - ...
- (Engineering) **Tasks:**
 - Setup GitHub project repo.
 - Setup GCP (or AWS) account, containers, and services.
 - (There might be Sub-Tasks for these too)
 - ...
 - Setup Jenkins CI pipeline.
 - Design overall (high-level) system architecture.
 - Research and decide on unit test and mocking framework.

NOTE: We will identify both sub tasks and engineering tasks together as Tasks.

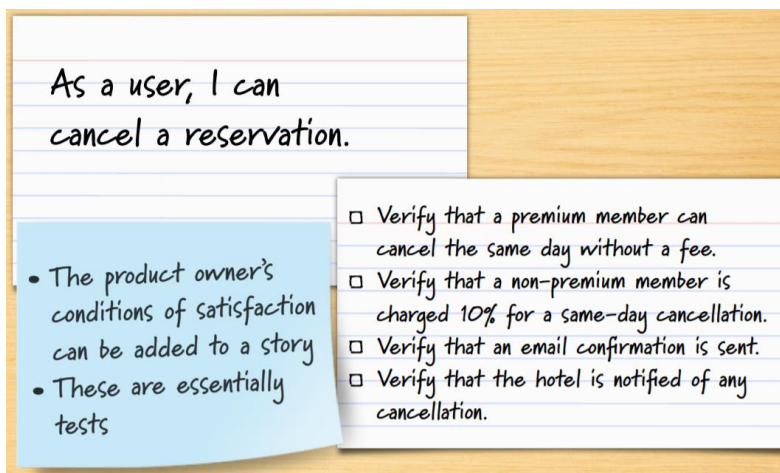
Guidelines for Good Stories

- Start with goal stories
- Write closed stories (stories that have a definite end point)
- “A recruiter can review resumes from applicants to one of her ads” instead of “A recruiter can manage the ads she has placed”
- Put constraints on the system on cards & implement automated tests for them
- Size your story appropriately for the time frame it may be implemented in
- Keep the UI out as long as possible
- Don’t rely solely on stories if somethings are better expressed in other ways
- Include user roles in stories rather than saying “user”
- Write for a single user (“A Job Seeker” not “Job Seekers”)
- Use Active Voice

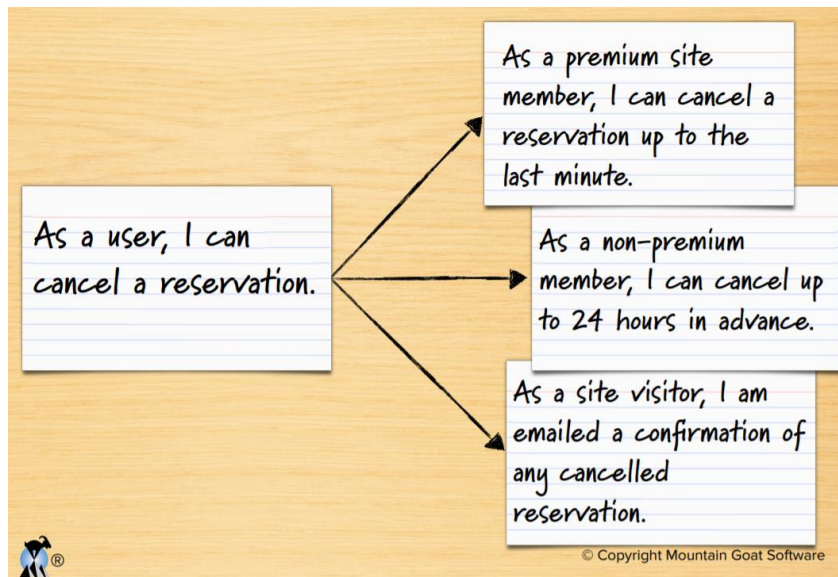
Example: Travel Web Site



- As a user, I can cancel a reservation
 - Does the user get a full or partial refund?
 - Is the refund to her credit card or is it site credit?
 - How far ahead must the reservation be cancelled?
 - Is that the same for all hotels?
 - For all site visitors? Can frequent travelers cancel later?
 - Is a confirmation provided to the user?
 - How?
- Details as conditions of satisfaction



-
- Details added in smaller sub-stories



Prioritize stories in a backlog

- Agile customers or product owner prioritize stories in a backlog
- A collection of stories for a software product is referred to as the product backlog
- The backlog is prioritized so that the most valuable items have the highest priorities

General guideline for setting priority – The MoSCoW method

The MoSCoW model, credited to pioneering data scientist Dai Clegg, has roots in Agile software development. The MoSCoW model has nothing to do with Russia's capital. Rather, it takes its name from the letters that make up its criteria: Must Have, Should Have, Could Have and Won't Have.

- **M-Must Have** — If you would have to cancel your release if you couldn't include it, then it's a Must Have. Must-Have user stories are those that you guarantee to deliver because you can't deliver without, or it would be illegal or unsafe without. If there's any way to deliver without it — a workaround, for example — then it should be downgraded to Should Have or Could Have. That doesn't mean it won't be delivered. It means that that delivery is not guaranteed.
- **S-Should Have** — Should Have features are important, but not absolutely vital to the success of your release. They can be painful to leave out, and may have an impact on your product, but they don't affect minimum viability of your product.
- **C-Could Have** — Could Have items are those that are wanted or desirable but are less important than a Should Have item. Leaving them out will cause less pain than a Should Have item.
- **W-Won't Have** — Won't Have user stories are those in which everyone has agreed not to deliver this time around. You may keep it in the backlog for later, if or when it becomes necessary to do so.

Estimation:

Estimates are essentially story points which say the amount of effort needed to get that work done. A guideline to be followed when estimating story points are:

Effort in hours	Story points
4	1
8	2
12	3
16	4
20	5
24	6
28	7
32	8
>32 hours effort means split that story in two smaller stories	

Recommended estimation

