Steps for making SRS

Eliciting Requirement
- Meeting, discussion and collaboration with clients
- Distinguish wants and needs
- Check if the requirements are realistic and feasible

Expressing Requirement
- Framing the requirements
- Use case diagram
- User Stories
- Story Board

Prioritizing Requirements

Analyzing Requirement
- Is the requirements are clear
- Is the requirement consistent with the problem
- Is the requirement described fairly (Less where less is needed and more where more is needed)
- Check possible conflicts
- Use a checklist here

Managing requirment
- Tracking and managing changes
- Update requirements

High	Must be done
Medium	Important but can be done later or satisfied in another way
Low	Improvements

Software Requirement Collecting Cheat-sheet

Expressing requirements

User stories

Front Side of the card:

As a …., I want to …, so that….

INVEST
- **Independent**
- **Negotiable**(Should focuses on important aspects of requirements, should not focus on technical details)
- **Valuable**
- **Estimatable** (Time require to implement)
- **Small** (implementable in a time period)
- **Testable** (Acceptance Test)

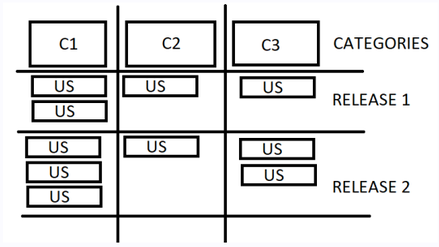
Backside of the card

Acceptance Tests:
- Criteria to meet the user story requirement
- Criteria of implementation to satisfy client

Product Backlog:
User stories with given unique number:
- Work Task (Not product feature but need to be done)
- Knowledge Task (Study)
- Bugs
- User Story

Prioritization:
- Must be done
- Should be done
- Could be done
- Won't be done

Story Map: 2D US map



Eliciting requirements

Questions to ask:

<u>If you want to elicit business requirements:</u>
Problem, Motivation, Highly successful solution
Successful solution worth,
Influence, Related projects,
Scope, Unintended consequences,
Difference between
Current solution

<u>If you want to to elicit business rules:</u>
Policy, Guideline, Law

<u>If you want to elicit user requirements:</u>
Goals, Expectation from the solution, aspects that excites you, most/least valuable aspects

<u>If you want to elicit non-functional requirements:</u>
Ask about different quality mentioned in non-functional requirements

<u>If you want to elicit external interfaces features:</u>
Events that the program will response, Environment
The product will use for, Exception condition of the Environment
Would anyone ever want to ...?
Could ... ever occur?
What should happen if ...?

After first meeting:
- **Use case diagram**
  - Name
  - Participating Actor
  - Goals
  - Triggers
  - Pre-conditions
  - Post conditions
  - Basic flow
  - Alternate flows
  - Exceptions
  - Qualities
- **Wireframes**
  - Doesn't contain any design element
  - Shows the location of ui elements
  - Doesn't specify color, pattern
  - Has to be done very quickly after initial meeting
  - Tool: Miro, Pencil, PowerPoint
- **Story board**
  - **Move style/Comic Strip**
    - Describes high level user experience with the product
  - **Glorified wireframes**
    - Shows the relation between different wire frames
    - Shows transition between states

Different types of Users

End User: The one who uses the product
Stakeholder: The one who is affected by the product by any means
- **Primary User:** End user/direct user of the product
- **Secondary User:** The one who will use the app occasionally or use it through and intermediary.

Tertiary User: Affected by the use of product or make discussion about the product

Human Limitations to consider

Perceptual/sensory Limitation: Limit in 5 senses of human (eg. Color Blindness)
Physical Limitation: Limit in human physical interaction (eg. Left Handed, handicapped)
Cognitive or Memory Limitation: Human memory limitation
Cultural Limitation: Cultural Background. (eg. Symbol, Lang.)

Types of Requirements

Business Rq.	Purpose/Goal of the project
Business Rule	Budget, Policy, Guideline, Regulation
User Rq.	Use cases, User stories, Story Boards, Scenarios
Functional Rq.	Input → Action → Output **Data flow diagram is used here**
Non-Functional Rq.	• Accuracy • Dependability • Usability • Security • Efficiency • Performance • Maintainability
External Interfaces	External API, Database etc **Data flow diagram is used here**
Physical product settings	Physical Environment
Development Constraints	Production Environment (Bandwidth, Processing Power, Memory, Platform, Environment)

Analyzing Requirement

Ambiguity causing words in user stories (Ignore them):
<u>Indirect:</u> SHOULD, COULD MAY, WILL
<u>Vague:</u> PROCESSED, HANDLED, OPERATED, ITEM, ENTITY, UNIT, AS APPROPRIATE, WHERE APPLICABLE, WITHIN REASON
<u>Completion:</u> AND SO ON, AND SO FORTH, etc, ALSO
<u>Persuasion:</u> CLEARLY, CERTAINLY, OBVIOUSLY
<u>Qualifiers:</u> ALL, EVERY, ONLY, NONE, NEVER, ALWAYS, SOMETIMES, OFTEN, USUALLY
<u>Comparative:</u> ...IS THE SAME AS…, -ER, -EST, ...IS MORE THAN…, ...IS AS…, ...AS
<u>Quantities:</u> A/ AN, SOME, MOST, FEW
<u>Pronouns:</u> HE/SHE, IT, WE, YOU, THEY, US, OUR, THIS/THAT
<u>Positional:</u> AFTER, BEFORE, FOLLOWING, LAST
<u>Temporal:</u> WHEN, FOR, UNTIL, FROM, CURRENT, LATEST
<u>Joining:</u> AND, OR, BOTH

Checklist for quality of User Story:
1) Correct (ACCURATE)
2) Complete (NO MISSING INFO)
3) Clear (NO AMBIGUITY)
4) Consistent (NO CONFLICT)
5) Feasible (POSSIBLE TO IMPLEMENT)
6) TRACEABLE (Trackable)

Software recommendations:
- Use case diagram: Libre draw
- Wireframes: InVission, Miro
- User Story and Story Map Trello, Azure

@princebillygk