AGILE USER STORIES
THE COMPLETE STORY

DAVID TZEMACH
WWW.DTVISIONTECH.COM
FEB 9 2016

WHAT IS A USER STORY…?

- Stories are added to the project "backlog" and from there to the "Sprint Backlog".
- After implementation, each story should add value to the overall effort.
- Every story should be visible and understandable to each team member.
- The story should be written based on the client perspective.
- A story is a basic description about what the customer wants to accomplish during the application development cycle.
- Each story provides an alternative vision for managing the requirements of the software.

USER STORY – STORY POINTS VS. TIME ESTIMATIONS

- Each user story will be estimated by "Story Points" instead of hours.
- Each team member have the power to affect the estimations by using is vote.
- The estimations are made by the scrum team members.
- The product owner is not part of the voting cycles.
- Story points can be translated into :
  - Shirt size (XS -> S -> M -> L -> XL -> XXL).
  - Fibonacci sequence (1 -> 2 -> 3 -> 5 -> 8 -> 13 -> 21)
  - Numeric numbers between 1-10

USER STORIES – THE RESPONSIBILITIES

	Product Owner	The Client	Scrum Master	Scrum Team
Who can Write stories..?	🟩	🟩	🟩	🟩
Who is the Owner of the story..?	🟩	🟥	🟥	🟥
Who should Maintain the story..?	🟩	🟥	🟥	🟥
Who should Prioritize the story..?	🟩	🟥	🟥	🟥
Who will Execute the story..?	🟥	🟥	🟥	🟩

THE BENEFITS OF USER STORIES

THE BENEFITS (1)

- The process of writing "User stories" is a great way to increase the collaboration between the team members.
- User stories will help to create a baseline of knowledge and expectations among the team members.

THE BENEFITS (2)

- User stories are great when you are working with agile methodology that empathies short iterations/sprints.
- User stories will help to determine the timelines and effort of each sprint.

THE BENEFITS (3)

- User stories will help to understand the scale of the project.
- The client describes the exact demands of the application.
- User stories will help the team members to monitor the project process.

HOW TO WRITE AN EFFECTIVE STORIES

THE GUIDELINES (1)

- The size of the story should be **small** enough in a way that it can be developed and tested in a single sprint.
- Every story should add **value** to the overall effort.
- A good story is the one that you can **estimate** (Timelines, Effort etc.).

THE GUIDELINES (2)

- Each story should be **independent** (dependencies may affect the prioritization and time estimations).
- The Story should be **flexible** to changes.
- The user story should be **testable**.

THE MISTAKES YOU CAN DO WHEN WRITING STORIES

THE MISTAKES YOU CAN DO (1)

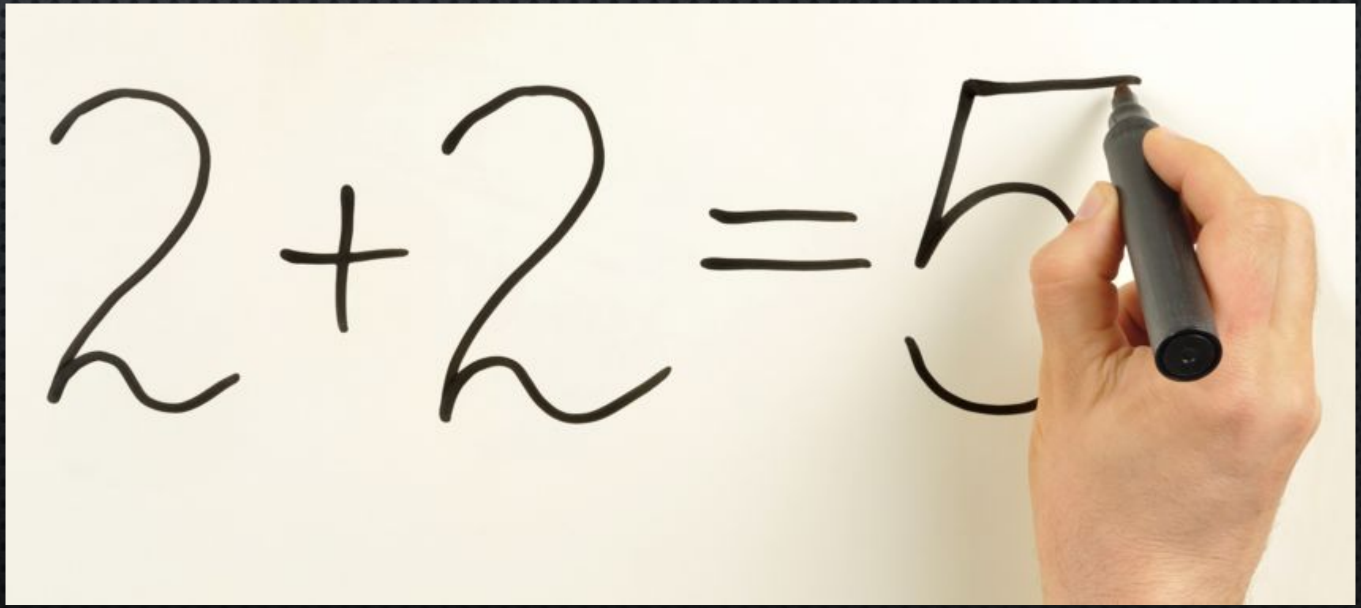
- Stories that are written without a preliminary conversation.
- Stories that are written from a technical perspective only.
- Too much detail on a single story (Keep it simple).

THE MISTAKES YOU CAN DO (2)

- Stories that doesn't contain the "Acceptance" criteria.
- Stories that doesn't contain the "Done" criteria.
- Stories that doesn't contain the requirements and specifications

THE MISTAKES YOU CAN DO (3)

- Stories that are too big to handle on a single sprint
- Stories that have too many dependencies
- Stories with high uncertainty



MY SUGGESTED TEMPLATE FOR WRITING USER STORIES

STORY TEMPLATE (TITLE)

- The title is built from max of 12 words, and should describe the main goal of the story.
- The title should be unique to this story so the scrum team can differentiate it from other stories that appear on the backlog.

STORY TEMPLATE (DESCRIPTION)

- The basic description can follow this template:

  As a <User>, I want <to achieve some goal> so that <I can accomplish…>.

- The description should fit to the index card.

ACCEPTANCE CRITERIA

What are the preliminary requirements that need to fulfill prior to the team can start to work on a story.

Examples:

- All bugs that affect this story are now fixed and verified.
- Dependencies on other tasks are now removed.
- The availability of Requirements

THE REQUIREMENTS FOR THIS STORY

Every story should include the requirements, that determines how the team should develop and test the story.

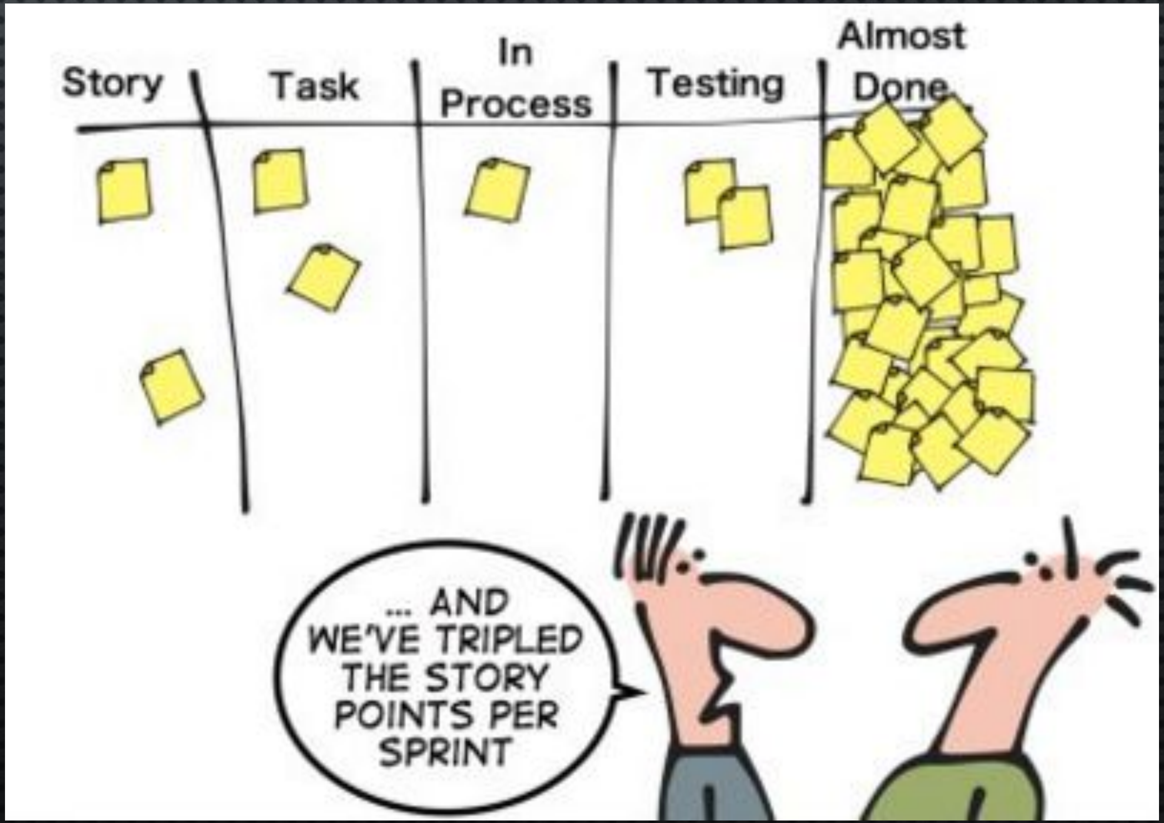
THE "DONE" CRITERIA

- What are the criteria that define if the team accomplished the story..?
- The "Done" criteria can be changed during the cycle (Based on the changed effort per story).
- A story can mark as "completed" only when the team accomplish this criteria.

EXAMPLES OF "DONE" CRITERIA

There are many different stories that you need to achieve during each sprint, this are few basic examples of what can be used as "Done" criteria:

- The functionality is ready for release.
- the code is covered by Unit tests.
- Design documents were created.
- There where No remaining bugs.
- Code review was done.
- The testing was done.
- Automation is ready.



FOR ADDITIONAL KB'S PLEASE VISIT MY BLOG

WWW.DTVISIONTECH.COM