User Stories

User Story Notes

SWEBOK Ch. 1 section 3.2

Elicitation Techniques -

User Stories:

- used in adaptive methods (like Agile)
- short, high-level descriptions of the required functionality
- “As a <role>, I want <goal/desire> so that <benefit>”
- just enough information to implement
- faster than detailed requirements statements
- need to consider how to determine if it was fulfilled.

SWEBOK Ch. 15 section 5.3

Prototyping -

- Use user stories to make a prototype
- During the prototyping stage of elicitation, new user stories will be created.

Constructing a prototype of a system is an initial version of the system to be constructed, it helps designers determine the feasibility of their design.

Uses for a prototype:

- Elicitation of requirements
- design and refinement of a user interface to the system
- validation of functional requirements

Prototyping is somewhat different between physical systems and software. Physical systems may be the first functional version of a system or they may be a model of the system.

In software engineering, prototypes are an abstract model of part of the software but are usually not constructed with all architectural performance, and other quality characteristics expected in the final product.

Prototype construction must have a clear purpose and be planned, monitored, and controlled.

29148-2018 6.3.3.3

- needs directly from the stakeholder(s)
- implicit stakeholder needs to be based on domain knowledge and context understanding.
- document gaps from previous activities
- prioritization necessary

Define stakeholder needs -

- Define the context of use within the concept of operations and the preliminary life cycle concepts.
  - ConOps describes an organization's assumptions or intent in regard to an operation or series of operations. The concept gives an overall picture of the organization’s operations, by providing a basis for bounding the operating space, system, capabilities, interfaces, and operating environment.
- Identify Stakeholder needs
  - Stakeholder needs include elicitation of needs directly from the stakeholder(s), identification of implicit stakeholder needs based on domain knowledge and context understanding, and documenting gaps from previous activities.
- Prioritize and down-select needs
  - Use the Decision Management process, Measurement process, System Analysis process, and Risk Management process to provide insight into the technical, cost, schedule, or other risks or opportunities.
- Define the stakeholder needs and rationale
  - Common sources and issues that need to be dealt with are:

- Goals – The term ‘Goal’ (sometimes called ‘business concern’ or ‘critical success factor’) refers to the overall, high-level objectives of the system. Goals provide the motivation for a system but are often vaguely formulated. It is important to assess the value (relative to priority) and cost of goals.
- Mission profile – How will the system perform its mission? How will the system contribute to the business or organizational operations?
- Operational scenarios - Scenarios can be used to define operational concepts and to bound the range of anticipated uses of system products, the intended operational environment, and interfacing systems, platforms, or products. Scenarios help identify requirements that might otherwise be overlooked.
- Operational environment and context of use - Requirements are derived from the environment in which the system or software product will operate. What are the conditions in which the system will operate in and how feasible are cost and design choices?
- Operational deployment – When will the system be used? Will it be deployed during the initial, middle, or wrap-up phases of a need?
- Performance – What are the critical system parameters to accomplish the mission?
- Effectiveness - How effective should the system be performing its mission? What are the applicable measures?
- Operational life cycle – How long will the system’s lifetime be?
- Organizational environment - Many systems are required to support an organization's process and this may be conditioned by the structure, culture, and internal politics of the organization. In general, new systems should not force unplanned changes to the business process.
- User and operator characteristics – Who will be using or operating the system? How will they vary in the role, skill level, and expected workload? What are the expectations or constraints on their capability and availability? Should allowance be made for accessibility?

5 Essential Agile techniques to Improve Your Requirements Documentation, especially about the INVEST principle

Source - https://reqtest.com/requirements-blog/5-essential-agile-techniques-to-improve-your-requirements-documentation/

1. Compliment User Stories with supporting artifacts - Jira allows attachments and links
    a. User stories that don't have enough detail, attach use cases, traditional requirements, or decision tables. This scenario may work well with situations of a client needing more documentation.
2. Create requirements that slice the cake - keep them small
    a. The best way to do this is by writing end-to-end user stories, think of slicing a multilayered cake where each layer represents a functional area of the product.
3. **INVEST** in your User Stories - (Independent, Negotiable, Valuable, Estimable, Small, and Testable)
    a. An ideal user story should have the following characteristics: Independent, Negotiable, Valuable, Estimable, Small, and Testable (INVEST). They should be as independent as possible. Avoid writing contracts, they should be negotiable. Make user stories valuable to the user and consumers. A story may not be estimable if the implementers lack domain knowledge or technical knowledge. This can be improved by writing effective user stories which follow the INVEST principles.
4. Groom your User Stories often
    a. Conduct User Story grooming workshops daily or weekly depending on how soon you need results. First, brainstorm stories on a whiteboard, then organize them into their user themes, prioritize the stories into high, medium and low and lastly improve high priority stories and make them follow INVEST principle.
5. Don’t be afraid to create prototypes
    a. It is better to spend a small amount of time building prototypes to just test out the waters, as it encourages bringing ideas to reality and encourages healthy discussions.

Recap

1. Ineffective Agile Requirements most often lead to failed products
2. The most common form of Agile requirements are User Stories
3. Compliment User stories with supporting artifacts, write well-rounded User Stories by slicing the cake, INVEST in User Stories, conduct User Story grooming sessions weekly or daily depending on your needs and create prototypes to complement your requirements
4. If these processes are new to your team you can start by talking to your team about the benefits of these processes and then introduce these processes gradually.

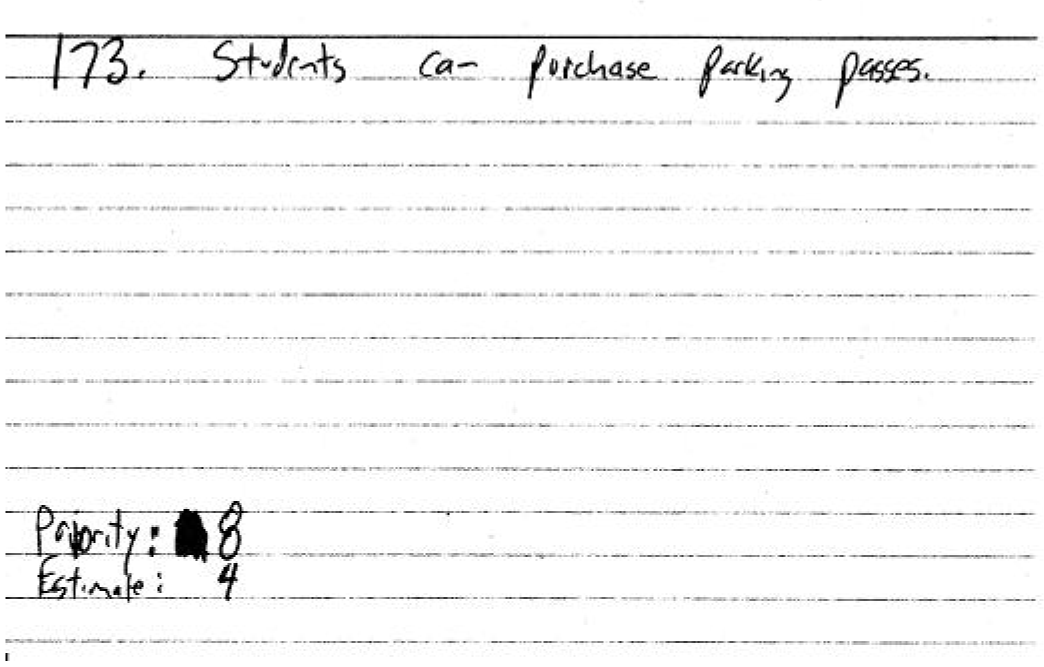
User Story agile modeling

Source - http://www.agilemodeling.com/artifacts/userStory.htm

- Reminder to have a conversation with your customer
- Stakeholders (can) write user stories

- Can be formal (follow template) or informal (doesn’t)

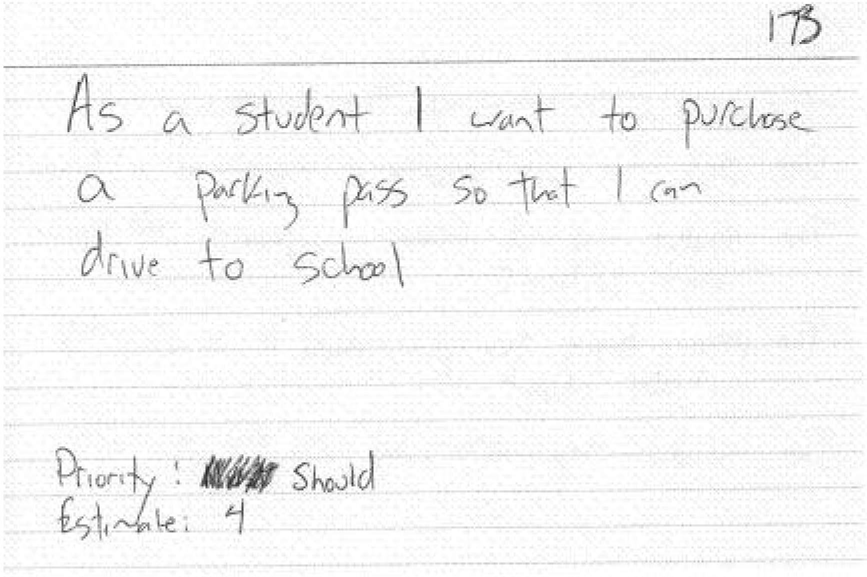
1. **Introduction to User Stories**
    a. A good way to think about a user story is that it is a reminder to have a conversation with your customer (in XP, project stakeholders are called customers), which is another way to say it's a reminder to do some just-in-time analysis. In short, user stories are very slim and high-level requirements artifacts.
2. **Initial User Stories (Informal)**
    a. User stories are small, much smaller than use cases and usage scenarios.
        i. Considerations when writing user stories:
            1. Stakeholders write user stories
            2. Use the simplest tool - Such as write user stories on index cards.
            3. Remember non-functional requirements - Stories should describe a wide variety of requirements types.
            4. Indicate the estimated size - Estimate the effort to implement the user story.
            5. Indicate the priority - Which is needed the most.
            6. Optionally include a unique identifier - A must for this class. Jira does it by itself.



173. Students can purchase parking passes.

Priority: 8
Estimate: 4

    b. (User story - Informal, high level)
3. **Initial User Stories (Formal)**
    a. Suggest formal format - As a (role), I want (something) so that (benefit).
        i. Book suggestion - “As a <role>, I want <goal/desire> so that <benefit>”



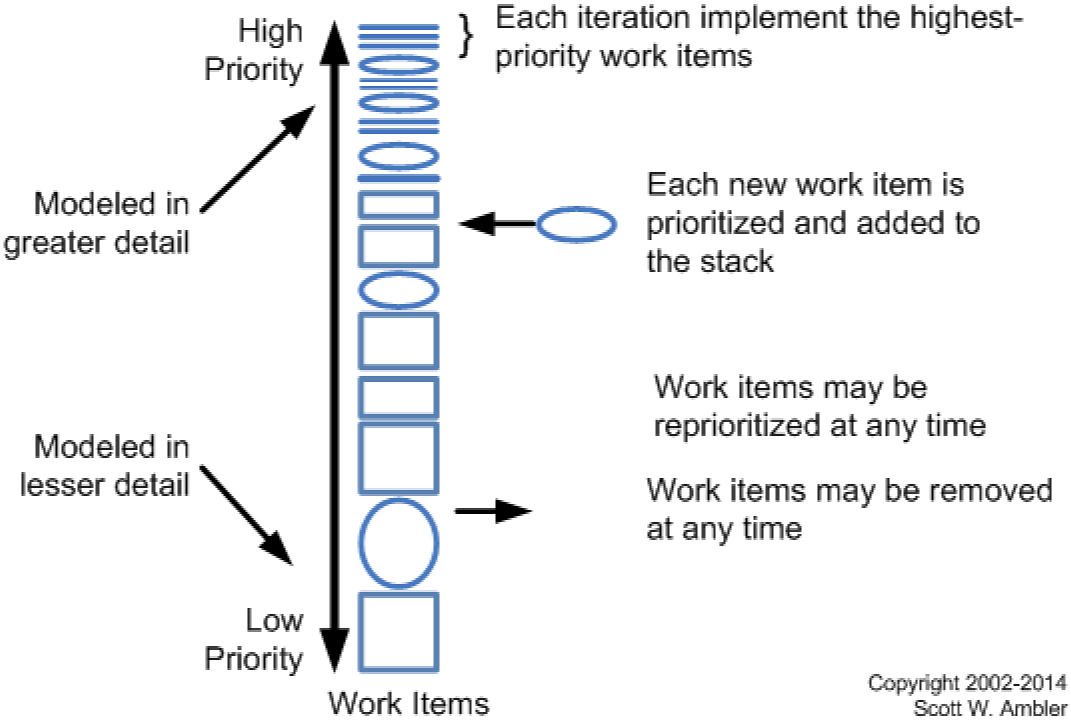
173

As a student I want to purchase a parking pass so that I can drive to school

Priority: Should
Estimate: 4

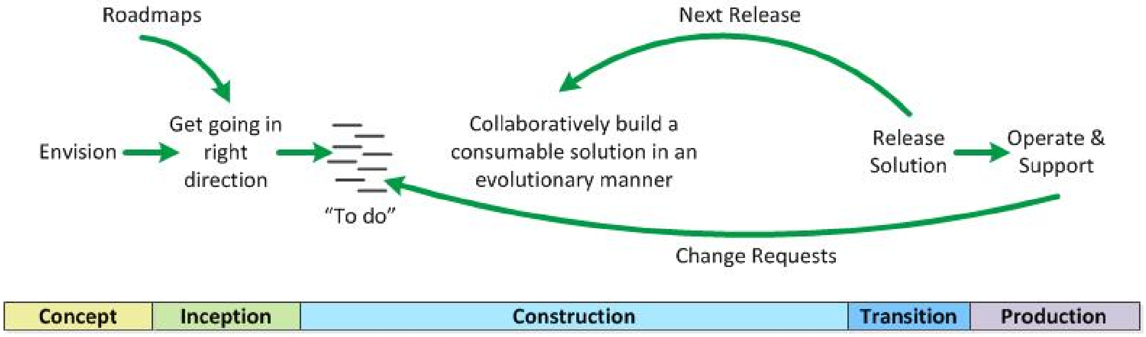
   b. (User story - formal, high level)

4. **User stories and Planning**
   a. Two areas where user stories affect the planning process on agile projects:
      i. Scheduling - Figure 4 depicts the agile change management process which is addressed in priority order.
      ii. Estimating


High Priority
Each iteration implement the highest-priority work items
Modeled in greater detail
Each new work item is prioritized and added to the stack
Work items may be reprioritized at any time
Modeled in lesser detail
Work items may be removed at any time
Low Priority
Work Items
Copyright 2002-2014 Scott W. Ambler

   b. Figure 4. The disciplined agile change management process

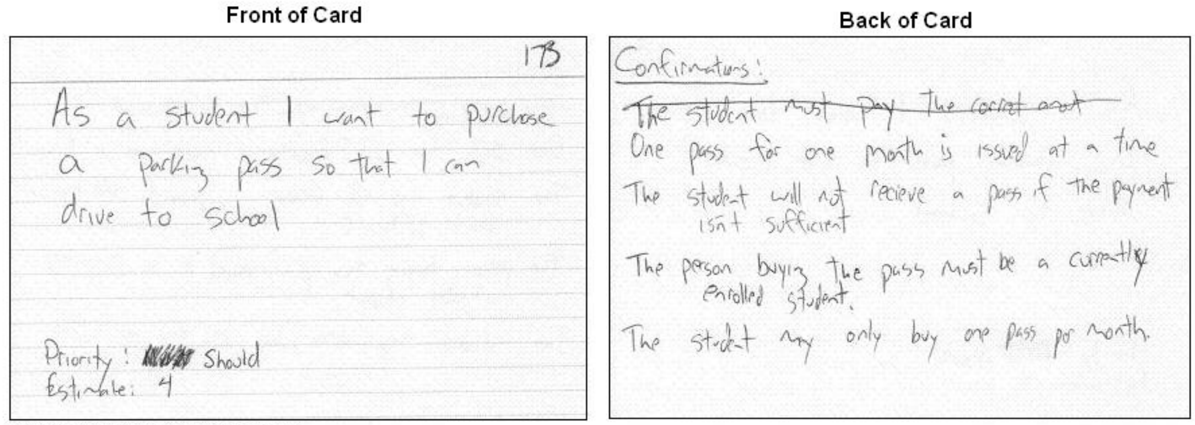
5. **User Stories Throughout the Agile Life Cycle**
   a. In the Disciplined Agile Delivery (DAD) life cycle of Figure 5, there are distinct “phases” in life cycle. The three common times when stories will be worked on during an agile project:
      i. Inception - Requirement envisioning activities to identify the scope of your system
      ii. Construction - The “to do” stage creating/validating user stories which could evolve over time making it iterative.
      iii. Transition - “End-Game” stage which could have new stories.



Copyright 2014 Disciplined Agile Consortium

   b. Figure 5. The Extended DAD lifecycle

6. **Detailing a User Story**
   a. User stories contain little information you will need to flesh them out a bit when you first work with them. There are three common times when you should do this
      i. During JIT analysis/model storming with stakeholders - Figure 7 shows how the backside of a user story can be used to capture the confirmations. Other tools could be used.
      ii. During iteration planning - Maybe common to list programming tasks required to implement a user story.
      iii. During implementation - Create rough sketches of what is going to be built, a flow chart or UML diagram could represent the relevant business logic.



Front of Card

173

As a student I want to purchase a parking pass so that I can drive to school

Priority: [redacted] Should
Estimate: 4

Back of Card

Confirmations:

~~The student must pay the correct amount~~

One pass for one month is issued at a time

The student will not recieve a pass if the payment isn't sufficient

The person buying the pass must be a currently enrolled student.

The student may only buy one pass per month.

Copyright 2005-2009 Scott W. Ambler

   b. (User story card - (formal, with confirmations)

7. **Epics**
   a. Large user stories, which are too large to implement in one single iteration there are broken down into smaller user stories.
   b. Epics are typically lower priority see Figure 4.

8. **Themes**
   a. A theme is a collection of related user stories. Themes are often used to organize stories into releases or to organize them so that various subteams can work on them.

User stories Atlassian

Source - https://www.atlassian.com/agile/project-management/user-stories

User stories are also the building blocks of larger agile frameworks like epics and initiatives.

- Stories keep the focus on the user.
- Stories enable collaboration.
- Stories drive creative solutions.
- Stories create momentum.

persona + need + purpose

What are agile user stories?

A user story is a small unit of work expressed from the software user’s perspective. The purpose is to articulate how a piece of work will deliver a particular value back to the customer. Stories fit nearly into agile frameworks like scrum and Kanban.

How to write user stories?

- Definition of “Done”
- Outline subtask or task
- User personas
- Ordered Steps
- Listen to feedback
- Time

Issues Tutorial Atlassian

Source - https://www.atlassian.com/agile/tutorials/issues

How to work with Issues in Jira

Key	Summary	Description	T	P
DB-20	As a Legal Counsel, I want the response bot to provide stakeholders with reliable answers to questions that would otherwise need to come from a lawyer so that more time-sensitive and critical issues could be dealt with.	As a Legal Counsel, I want a type of software that provides stakeholders with reliable answers to questions that would otherwise need to come from a lawyer – and take time away from more time-sensitive and critical issues – that could create some efficiencies and make a meaningful impact on the company.	🟩	↑
DB-19	As a Account Executive, I want a tool for quick legal information so that the legal team can focus on more substantive and critical issues.	As a Account Executive, Any tool to get me information more quickly would be beneficial, both to me and to the legal team, so that they could focus on more substantive and critical issues the company faces.	🟩	↑
DB-18	As a Sales Engineer, I want quick access to legal information so that I don't need to communicate with the legal team in person.	As a Sales Engineer, I want some real value in having quick access to information that we otherwise need to obtain from a lawyer, in person.	🟩	↑
DB-17	As a Account Receivable, I want relatively available legal information to access and navigate on my own so that the business process is more efficient.	As a Account Receivable, I want information we need available through a software tool we could access and navigate on our own, which would create a ton of efficiencies.	🟩	↑
DB-16	As a Sales member, I want quick access to legal answers so that I don't feel like I'm disturbing the Boomi legal team.	As a Sales team member, It would be helpful to have certain information readily available in a self-help tool, both so that I can get quick access to the answers, but also so that I don’t feel like I’m bothering the Boomi legal team.	🟩	↑

5 issues

User Stories in General - Can be an informal/formal explanation of a software feature written from the perspective of the end-user or customer. The purpose is to articulate how a piece of work will deliver a particular value back to the customer.