In this article What is a User Story? Who Writes User Stories? When are User Stories Written? Why Create User Stories? Types of User Stories Examples of User Stories User Story Templates Conclusion In this article What is a User Story? Who Writes User Stories? When are User Stories Written? Why Create User Stories? Types of User Stories Examples of User Stories User Story Templates Conclusion View All In this article, you will learn about User Stories, 3 C's of a user story, who writes it, how to write it, how to INVEST in user stories, different types of user stories with examples and acceptance criteria. Also, read more about Power BI Developer Future. What is a User Story? User Story is a tool in which requirements are captured in an easy to understand plain language, and is written from the perspective of an end user. "In software development and product management, a user story is an informal, natural language description of one or more features of a software system. User stories are often written from the perspective of an end user or user of a system" In Agile software development, user stories are used to express the requirements from an end user perspective. The format of the user story is: As a < user > I want to < perform an action > So that < I expect…. > Copy Code - is the end-user or the role of the user in the application software – "As a net banking customer" - the action the user is performing on the application software – "I want to add beneficiary in my account" - outcome, desired value, the user expects out of the action performed – "so that I can transfer money to the added beneficiary" The larger-sized stories are called "Epics" which are then decomposed to "Features" and then further decomposed to a "User Story". Epic example: As a Bank, I want to provide net banking to customers, so that they can perform various transactions. The above Epic can then be decomposed into multiple features: few examples: As a Bank, I want to provide a funds transfer feature to customers, so that they can transfer funds from one account to another account As a Bank, I want to provide an account summary for all the customer's types of accounts. As a Bank, I want to provide credit card details to customers. Now each feature can be decomposed further into multiple user stories. User stories, based on the estimated size, are taken for implementation in an iteration. User stories should be granular enough that they can be completed within an iteration and cannot be continued in the following iteration. If a story cannot be completed within an iteration, the same should be split logically. User stories are prioritized by the product owner based on business priority and are available at the top of the product backlog. The dev team pulls the stories into an iteration backlog and implements them. The Definition of Done(DOD) for user stories is decided by the team which includes acceptance criteria, and processes that need to be followed like unit testing, regression testing, code review, etc. The story is said to be "done" only when it meets the preset Definition of Done. Know more about agile vs traditional project management. Who Writes User Stories? So, whose responsibility is to write user stories in an agile team? Generally, the notion is that only the Product Owners should write user stories as they are the ones who elicit requirements from the stakeholders. However, in practice, any member of an Agile team may write user stories, though the overall responsibility is that of a Product Owner. The product owner should go through the stories and prioritize them in the product backlog. Over the course of an agile project, every team member is encouraged and expected to write user stories. Prepare for your interviews with these top Microservices interview questions. When are User Stories Written? Are user stories written at the beginning of the project in a traditional way? User stories are written throughout the lifecycle of the project. At the start of the project, user stories are written in Sprint '0', also called as pre-sprint. Initially, the product owner elicits the requirements from the stakeholder and they are stored as EPICS, Features and User Stories in the product backlog. The requirements in agile software development are progressively elaborated and hence the need for writing user stories will arise throughout the project. These are written mainly during the backlog grooming session where the product owner decomposes epics/features into granular stories. The Dev team writes stories along with the product owner during this session and also gets involved in the 3 C's (the next section describes this). confirmation in the 3C's of user stories "Card", "Conversation" and "Confirmation" is a model that captures the components of a user story. This is popularly known as the 3Cs model which helps in planning and estimating the user stories by the Agile team. "Card" – denotes a Post-It note or physical card, typically 3"x5" in size, where the important information of a user story is captured. The card should contain enough information (not too less or too much) that the team is able to understand in order to plan & estimate the story. "Conversation" – this is the conversation that happens between the product owner and the dev team to discuss the story and get into the details. This may also be a conversation with the users of the system. This conversation also brings out the creativity of the dev team and uncovers any unstated needs of the users. "Confirmation" – this brings out the acceptance criteria for a story based on the above conversation. This criterion will be used to evaluate the story by the stakeholders when the user story is implemented by the dev team. The 3 C's of the user story generally unfold during the backlog grooming session when the dev team and the product owner discuss the stories that need to be groomed. The user stories are written during this time as well on the card by the dev team and product owner. Just enough information is captured in the story that enables the team to discuss and get into the details, uncovering any hidden or explicit information in the process. The team then negotiates with the product owner and arrives at the acceptance criteria for the user story. Next, the dev team estimates the user story with the available information. The conversation continues between the dev team and product owner until a consensus is reached with respect to the details and acceptance criteria and until the team can size the same. This round of conversation may happen again during the iteration/sprint planning session. The dev team then implements the story in an iteration which is reviewed by the product owner or stakeholders at the end of the iteration. They will then accept the story based on the acceptance criteria defined for the story. You can also, check out the blog post on how Agile teams can use the 5 whys root cause analysis process to identify the main reason for the problem. Why Create User Stories? What are the benefits that a team will get by documenting the need of the stakeholders in the form of user stories? It enables the team to understand the requirements from a user perspective. The focus is on the user to provide value to them; the user story clearly describes the expected outcome of every action performed. This manner of capturing requirements provides opportunities for the team to collaborate more with the product owner and business users. By having conversations (in 3 Cs), the team is able to uncover the hidden requirements and also come up with creative solutions. Provides a shared understanding of the requirements to the team so that everyone is aware of the outcome/goal of the story and is on the same page. User stories help the team to achieve smaller product increments. User stories are more understandable by all stakeholders (technical/non-technical/business/operations). User stories help the team to implement features in smaller iterations ranging from one week to one-month durations. User stories enable the team for progressive elaboration, where they can defer the story until more clarity is obtained. User stories help create transparency of the priorities defined by the product owner and the customer. User stories help the developers, product owner and business owners to reach a mutual consensus as they discuss the details and agree on the acceptance criteria. With the help of the user stories tool and CSM or PMP course, professionals can enhance their career skills in

agile project management. This helps prioritize the product features by the stakeholders and also helps to take the right decisions at the right time. Types of User Stories We can classify user stories into functional and technical types: Functional: Normally, a user story is written based on the functional aspects of the application software, while focusing on the user and the value of the functionality provided to the user. Functional stories concentrate on the product features which the customer will be testing at the end of an iteration based on the acceptance criteria defined for the story. Technical: Technical stories are written to be able to support the functional stories. Technical stories can be classified as Infrastructure stories: any infrastructure addition/modification that may be required to support the functional story Refactoring: this type of story is written to maintain the code and address technical debts. This can be used for designing and automation needs as well Spikes: stories that require research on architecture and design which may in turn help achieve the functional need of the customer. Examples of User Stories Let us see some examples of user stories (Epics, Features and User Story) in this section. IDÂ EPICSÂ E1Â As a Sales Professional, I want to generate reports so that I can take a decision on the marketing strategy for the upcoming quarter E2Â As a Banking Customer, I want to access net banking, so that I can access my account and make transactions E3Â As an Administrator of the software, I want to access master records so that I can make changes to customer data IDÂ Features E2F1Â As a Banking Customer, I want to access Savings account so that I can view/make transactions E2F2Â As a Banking Customer, I want to access Credit Card page, so that I can view and make transactions E2F3Â As a Banking Customer, I want to access Loans page so that I can view my loans E2F4Â As a Banking Customer, I want to transfer funds, so that I can move my funds to different accounts within my bank and other banks ID User Stories E2F1U1 As a Banking Customer, I want to access/view summary of my savings account, so that I know my balance and other details E2F2U1 As a Banking Customer, I want to Login to Net banking so that I can view credit card details E2F4U1 As a Banking Customer, I want to transfer funds within my own accounts so that I can move some balance across my accounts E2F4U2 As a Banking Customer, I want to transfer funds from my account to another account in another bank, so that I can send money to my family and friends who have accounts in other banks E2F4U3 As a Banking Customer, I want to add beneficiary to my account, so that I can transfer funds to the beneficiary Technical Stories E2TU1 As a Net Banking Administrator, I want to have the customer's data backed up so that I can restore it any time in case of issues E2TU2 As a Net Banking application, I want to shake hands with another bank using a specific formatted XML so that funds can be transferred based on the customers' needs User Story Templates You can utilize four user story templates available for download to capture new product functionality from the user's perspective and specify your acceptance criteria. Each template presents a distinctive layout that varies in the level of detail you wish to incorporate. Simple user story template Epic user story template Thematic user story template SAFe user story template INVEST in User Stories This is an acronym for a set of attributes or criteria that helps us to assess the quality of the user story. If any of the attributes falls short in a story, it means that the team may want to consider rewriting the user story. 1. Independent: User stories should be independent of other stories. There should be no overlap between them. They can however follow one after the other in a sequence, in a way that makes it easy to schedule and implement them. Â This is one of the challenges that the team faces especially when they have just started adapting agile ways of working. They may have a story which is dependent on something else which may be done by another team. The teams may hope that they can run the two stories in parallel and by the time the first team is done, the dependent team will also complete their part of the story. Â This is not the right way of running user stories, as it can result in a lot of confusion and blame. The advantage of having independent stories is that there is no blame game across teams. It also allows to consider the dependencies and come up with innovative ways of removing them to become independent. 2. Negotiable: The story should not be written in so much detail that it becomes a requirement document. If it is in too much detail, it does not give an opportunity for the dev team to have any conversation with the product owner or the business. The story should be written with just enough detail so that it paves the way to open discussions with the product owner or business, and helps to elicit details or come up with creative solutions. By negotiating on the story with the relevant stakeholders, teams can come to a common understanding. 3. Valuable: The story should be valuable to the customer. It should clearly state why we are doing this? How is it going to produce value to the customer? What value will the customer realize by implementing this story? Â The only reason why user stories should be part of the product backlog is that they add value to the customer, right? 4. Estimable: The user stories should have sufficient detail for the dev team to understand and estimate them. The conversation in 3 C's helps the team to uncover the details with the product owner and stakeholders, so that they can size the story. If the story is too big and not sizeable, then the story should be refined or decomposed further. Whatever information the team may require should be available in the story for them to estimate it. In case there is a part of the story where the team has to do research, then a "spike" story may be created while the rest of the story can be estimated and taken for implementation. 5. Small: Good user stories should be small. This does not refer to the size or number of words written in a story. A small story is of the right length so that the implementation team can complete the story within an iteration. It should be small enough that the story is "fully delivered" during an iteration. A small user story helps the team to develop and test quickly and easily. 6. Testable: A good user story should be testable in order to be "Done". This is supported by the "Confirmation" in 3 C's where the team comes up with acceptance criteria for every story after the detailed conversation with the stakeholders. The customer should be clear about what he should test during the review. If he is not clear, then the story is not good enough to be implemented. The team works together in a collaborative way to INVEST in good stories. The team learns to write good user stories as they work together and also proactively think about the values and criteria that are laid out in INVEST. Conclusion Transformation of documentation on user requirements in a Functional Requirements Document (FRD) or Software Requirement Specification (SRS) in a traditional project management, towards User Stories in Agile project management, is a massive step. It helps shift the mindset of how teams can understand and collaborate with the customer in a better way, by shifting their focus of implementation towards value that the customer may realize from the story. This shift has worked very well in terms of meeting the requirements and expectations of the customer. Krishnakumar Kuppusamy Author Krishnakumar Kuppusamy is one of the highly experienced Agile Coaches and SAFe Program Consultant (SPC 5.0). He has 24+ years of experience in information technology industry handling both traditional and agile projects. He has worked with companies like Citibank (USA) & Polaris Software at various capacities in project & program management. He has worked for ANZ and Ford India, coaching multiple Agile teams in their transformational journey. He is also a freelance trainer, conducted trainings in SAFe/PMP/PMI-ACP/ITIL/CBAP for over 2000+ professionals helping them getting certified and excel in their respective areas. Share This Article Ready to Master the Skills that Drive Your Career? Avail your free 1:1 mentorship session. First name* Last name* Email* +49 Phone* Your Message (Optional) I consent to receive communications by email, SMS, phone and other electronic means from KnowledgeHut. See our privacy policy and terms & conditions for your rights under GDPR. Talk to Us Upcoming Agile Management Batches & Dates Name Date Fee Know more Certified ScrumMaster (CSM)Â® Cohort starts on 12 Aug 2023, Weekend batch EUR 595 View Details Certified Scrum Product OwnerÂ®(CSPO) Certification Cohort starts on