SAMSUNG SW ACADEMY FOR YOUTH

對外秘

프로젝트 명세서

사용자 스토리(User Story) 작성

對外秘

목차

1. 프로젝트 개요

본 과제는 사용자 스토리에 대해 학습하고 작성을 연습해보는 과제입니다. 이 과제를 통해서 교육생은 사용자 스토리를 작성해보고 동료와 소통하며 요구사항을 정리하는 방법을 배우게 합니다.

사용자 스토리(User Story)는 요구사항을 정의하기 위한 한 방법으로서 주로 애자일(Agile)과 XP(eXtreme Programming) 방법론에서 많이 활용합니다. SSAFY 에서는 Jira 를 사용해 애자일 방법론 하에서 프로젝트를 진행합니다. Jira 의 일감 유형 중 이야기(Story)와 큰틀(Epic)을 사용자 스토리 형태로 작성해 관리하는 방법도 있습니다. 본 과제를 통해 사용자 스토리 작성을 연습하고 익힌 내용을 2 학기 본 프로젝트에도 활용해 보시기 바랍니다.

2. 사용자 스토리 개요

사용자 스토리는 요구사항을 기술(describe)하는 방법의 하나로 주로 아래와 같이 문장 형태로 작성합니다.

- 나는 <사용자 유형>의 입장에서, <구체적인 이유>를 위한 <목표>를 원한다.
  - 예시) <교육생>은 <수강 신청>을 위해 <과목의 목록을 볼> 수 있다.

사용자 스토리는 일반적으로 아래의 것들을 포함합니다.

- 서술(또는 Card) : 위와 같은 문장.
- 대화(Conversation) : 세부사항은 <대화>를 통해 구체화 합니다.
- 테스트(또는 Confirmation) : 세부사항을 문서화한 것으로, 스토리의 완료 조건을 설명합니다.
  - 예시) 과목 목록에는 과목명, 담당교수(강사), 강의실이 표시되어야 한다.

서술은 사용자(또는 고객)에게 어떤 목적으로 무엇이 필요한지 설명합니다. 개발에 앞서 개발 담당자와 사용자는 대화를 통해 세부사항을 결정하게 됩니다. 결정된 세부사항과 예상되는 동작을 테스트에 작성하며, 이를 통해 개발 담당자는 해당 스토리의 완료를 판단할 수 있습니다.

사용자 스토리에는 기존의 요구사항 정의 방법과 비교해 다음과 같은 특징이 있습니다.

- 사용자 중심 : 사용자를 중심에 두며 사용자(또는 고객)가 직접 작성하기도 합니다. 기술 용어가 아닌 비즈니스 언어로 작성하여 사용자와 개발자 모두가 이해할 수 있습니다. 사용자를 직접 만나기 어려운 경우 Project Manager, Product Owner, 제품 기획자 등이 작성을 주도하기도 합니다.
- 사용자 입장에서 서술 : 특정 사용자(Persona)를 기준으로 작성합니다. 기존 요구사항 정의 방법에서는 일반적으로 서비스를 제공하는 **시스템의 입장에서** 서술했기 때문에 알려지지 않은 사용자, 권한 문제 등을 고려하기 어려웠습니다.

사용자 스토리는 반대로 사용자의 입장에서 기능을 서술함으로써 위와 같은 문제가 줄어들게 할 수 있습니다. 또한 개발자(제작자)가 사용자의 관점에서 생각하도록 유도하여 사용자 경험을 높이는 장점도 있습니다.

- 대화를 위한 시작점의 역할 : 스토리는 **세부사항을 포함하지 않으며** 기능을 논의(대화)하기 위한 단서로서의 역할을 합니다. 애자일과 XP 에서는 문서를 통한 소통 보다는 사용자와 직접 대화하는 것을 중요하게 여깁니다. 테스트 부분도 대화를 유발하는 기능을 합니다. 개발 담당자가 테스트에 기록된 내용 중 불명확하거나 독특한 정보가 있다면 사용자에게 확인을 하게 됩니다.

처음부터 스토리를 잘 작성하기는 쉽지 않을 것입니다. 스토리를 잘 작성하기 위해서는 아래와 같은 지침들을 따라 보시기 바랍니다.

- 사용자가 시스템(또는 서비스)을 사용하는 <주 목적>을 중심으로 작성하라.
- 한 명의 사용자(Persona)에 대해서 작성하라.
- 닫힌 스토리로 작성하라.
- 하나의 스토리가 시작부터 끝까지 기능을 제공하도록 작성하라.
- 능동태로 작성하라.
- 제약사항을 기록하라.
- 가까운 시기에 개발할 기능은 작은(디테일한) 스토리로, 나중에 개발할 기능은 큰(추상적인) 스토리로 작성하라.

사용자 스토리를 기반으로 요구사항을 정의함으로써 다음과 같은 효과를 기대할 수 있습니다.

- 실제로 필요한 시점(개발을 시작하게 되는 때)까지 세부사항을 정하는 것을 미룰 수 있게 되며 전체 요구사항을 파악해야 하는 비용을 줄일 수 있습니다.

- 사용자가 가장 필요로 하는 기능부터 개발할 수 있게 됩니다.
- 변경과 변화에 용이합니다.

폭포수(Waterfall)와 같은 전통 프로세스에서는 초기에 전체 요구사항을 명확히 정의하고 다음 단계로 넘어가게 됩니다. 이로 인해 테스트 단계에서 발견되는 요구사항의 변경, 설계 오류 등을 반영하기 어려웠습니다. 사용자가 필요로 하지 않는 기능에 대해서 정의하고 설계하는 낭비가 있기도 했습니다. 또한 개발 이전의 사전 작업에 많은 시간이 필요하므로 서비스의 제공이 늦어졌습니다. 사용자 스토리를 사용하면 고객에게 가장 높은 가치를 제공하는 것부터, 빠른 시점에 제공할 수 있게 됩니다.

사용자 스토리는 주로 <기능 요구사항>에 대해 작성하지만 <비기능 요구사항>도 일종의 제약사항으로 볼 수 있으며 이에 대해서도 다룰 수 있습니다. 예를 들어 회원가입과 관련된 사용자 스토리에 '사용자 정보는 암호화되어야 한다'라는 제약사항을 테스트 내용으로 기록할 수 있습니다. 추가로 사용자 스토리로 포함하기 어려운 기술적 내용, 관리 업무, 인프라 개선 등의 작업은 기술 스토리(technical story)로 사용자 스토리와 구분해 작성하기도 합니다.

작성된 스토리는 스토리 포인트(Story Point)를 부여받고 제품 백로그(Backlog)에 들어가게 됩니다. 이후 릴리즈 계획, 스프린트 계획 등을 거쳐 백로그의 스토리들은 스프린트 백로그로 옮겨지고 스프린트를 통해 개발 결과물로 발전해 갑니다.

3. 참고 자료

- 사용자 스토리

  https://hanminwoo.com/75?category=685688

- 유저스토리 모르는 사람 없게 해 주세요.

  https://brunch.co.kr/@workingus/36

- 사용자 스토리 포인트로 스마트하게 프로젝트 진행하기(Line Engineering Blog)

  https://engineering.linecorp.com/ko/blog/user-story-point-in-line-pay-team/

4. 과제

본 과제는 별도의 툴 없이 진행할 수 있으며 결과물을 문서 형태로 정리해 제출합니다.

1. 주제 선정

◆ 본인이 많이 사용하는 서비스 중 하나를 선택합니다.

- Ex) 배달 앱(배민, 요기요), SNS(트위터, 인스타그램), 쇼핑몰, 영화 예매, 철도 예매 등

2. 역기획으로 사용자 스토리를 작성

◆ 서비스의 일부 과정에 대해서 사용자 스토리를 작성해 봅니다.

- 참고) 배달 앱에서 음식점을 검색하고 주문하고 배달이 완료되기까지의 과정

◆ 하나 이상의 사용자 유형이 있으면 다른 사용자에 대해서도 스토리를 작성해 봅니다.

- 참고) 배달 앱의 경우 소비자(배달 요청)와 공급자(음식 제공), 배달 담당자를 고려해볼 수 있습니다.

◆ 스토리 작성은 인덱스 카드, 포스트잇 등 물리적인 도구를 사용해 작성하거나 Canva 의 온라인 화이트보드, Figma, Miro 등 온라인 협업 도구를 사용해 작성할 수 있습니다.

3. Peer Review(동료 검토)

◆ 작성한 스토리를 동료와 함께 검토해 봅니다. 아래와 같은 기준을 사용해 검토를 할 수 있습니다.

- 검토 기준 : 좋은 스토리의 여섯 가지 특징(INVEST)

  ■ 독립적이다(Independent) : 스토리 간에 **의존성**이 있으면 추정/우선순위 선정 등에 문제가 있을 수 있습니다. 따라서 사용자 스토리는 가급적 독립적으로 작성해야 합니다.

  ■ 협상 가능하다(Negotiable) : 사용자 스토리는 계약이 아니므로 세부 내용에 대해서는 협상(조정) 가능해야 합니다. 이러한 협상(조정)은 실제 작업을 계획할 시점에 이루어집니다.

  ■ 사용자 및 고객에게 가치가 있다(Valuable) : 사용자에게 가장 가치 있는 기능부터 개발할 수 있기 위해서는 사용자가 스토리를 보고 가치를 판단할 수 있어야 합니다.

  ■ 추정 가능하다(Estimatable) : 스토리는 개발팀에서 규모를 추정할 수 있어야 합니다. 추정치는 일반적으로 스토리 점수(Story point)가 이용됩니다. 스토리 점수를 산정하는 데에는 플래닝 포커, 피보나치 수열, 티셔츠 사이즈 기법 등을 사용할 수 있습니다.

  ■ 작다(Small) : 스토리가 너무 크면 추정이 어려워집니다. 따라서 한두명의 개발자가 반나절에서 한 번의 스프린트(1~2 주일) 안에 구현하고 테스트할 수 있는 크기가 적당합니다. 너무 큰 경우 에픽(Epic, Jira 에서는 큰틀)으로 분류하고 세부 스토리로 분할하거나 작은 스토리로 분리합니다. 반대로 너무 작은 경우 다른 스토리와 합치는 것을 고려할 수 있습니다.

  ■ 테스트 가능하다(Testable) : 스토리가 너무 개념적으로 작성되면 해석의 여지가 많아지고 오해가 생기며 완료 시점을 알기도 어려워집니다. 따라서 스토리는 테스트 가능한 수준으로 기술되어야 합니다.

◆ 검토 내용을 정리하여 사용자 스토리를 보완합니다.

5. 산출물 제출

https://lab.ssafy.com/s10-study/self-project/ 의 “산출물 제출 가이드.docx” 참조