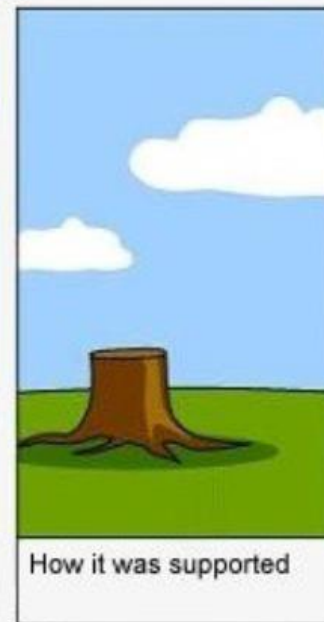
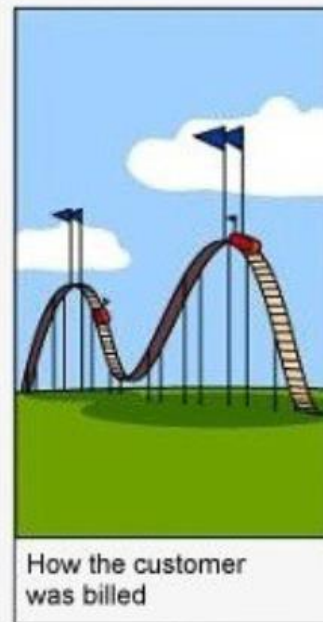
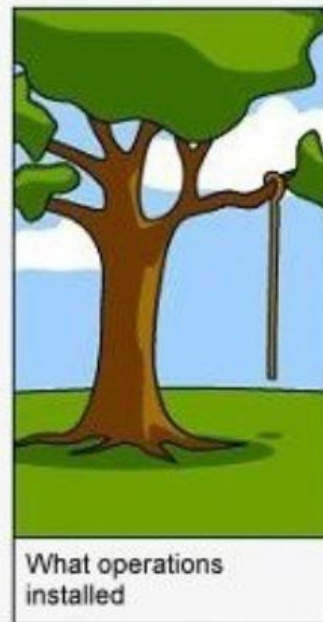
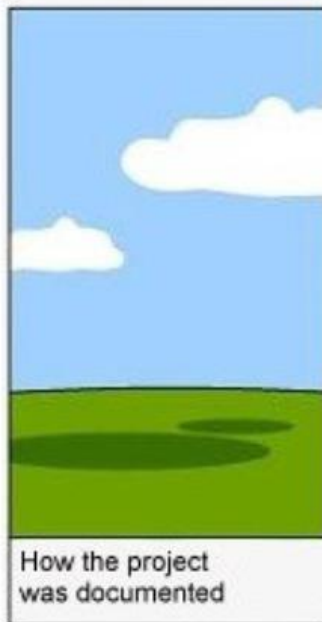
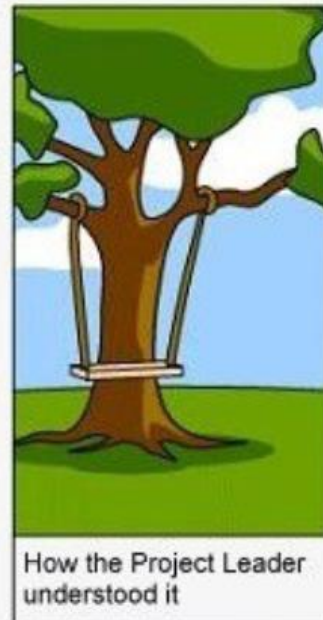
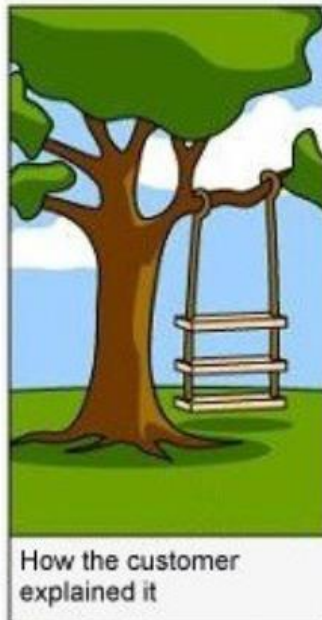




Requirements

For External
Courses

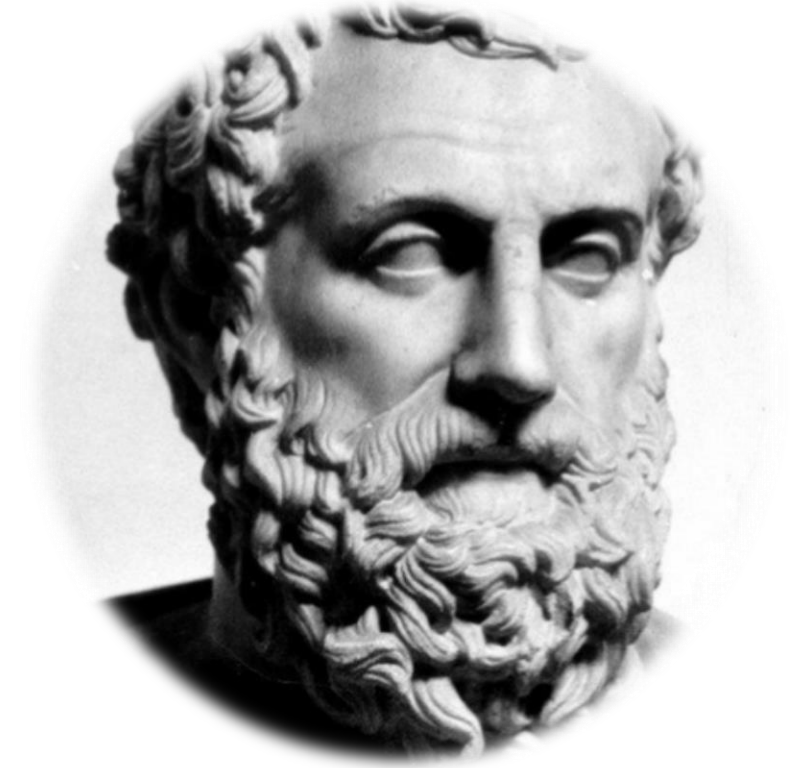
2022



Criteria

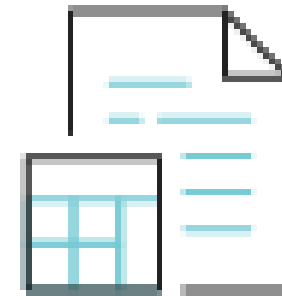
Acceptance criteria - the criteria that a component or system must satisfy in order to be accepted by a user, customer, or other authorized entity.

Entry criteria - the set of conditions for officially starting a defined task.



Requirement

A provision that contains criteria to be fulfilled.





The issue with this methodology is that if a project has a big scope, it might take many months if not years until we get final results. A lot of things can change and go the wrong way. Sometimes during the final stage of acceptance and testing, we might discover that the original design stage was wrong. The success rate of a big project executed in this way is alarmingly low

Types of Requirements

Business requirements

User requirements

Business rules

Quality attributes

Functional requirements

Non-functional requirements

Limitations, constraints

External interfaces requirements

Data requirements

Software requirements specification

Quality Attributes: Important to Users

| | |
|--------------------------|--|
| Availability: | Is it available when and where I need to use it? |
| Installability: | How easy is it to correctly install the product? |
| Integrity: | Does it protect against unauthorized access and data loss? |
| Interoperability: | How easily does it interconnect with other systems? |
| Performance: | How fast does it respond or execute? |
| Reliability: | How long does it run before experiencing a failure? |
| Recoverability: | How quickly can the user recover from a failure? |
| Robustness: | How well does it respond to unexpected operating conditions? |
| Safety: | How well does it protect against injury or damage? |
| Usability: | How easy is it for people to learn and use? |

Quality Attributes: Important to Developers

| | |
|-------------------------|--|
| Efficiency: | How well does it utilize processor capacity, disk space, memory, bandwidth, and other resources? |
| Flexibility: | How easily can it be updated with new functionality? |
| Maintainability: | How easy is it to correct defects or make changes? |
| Portability: | How easily can it be made to work on other platforms? |
| Reusability: | How easily can we use components in other systems? |
| Scalability: | How easily can I add more users, servers, or other extensions? |
| Supportability: | How easy will it be to support after installation? |
| Testability: | Can I verify that it was implemented correctly? |

Good requirements should have the following characteristics:

| | | | |
|---------------|----------------|-------------|--------------|
| Completeness | Atomicity | Consistency | Clearness |
| Feasibility | Obligatoriness | Up-to-date | Traceability |
| Modifiability | Priority | Correctness | Testable |

Static testing

Testing a work product without the work product code being executed.

Static testing

Static test techniques provide a powerful way to improve the quality and productivity of software development.

The fundamental objective of static testing is to improve the quality of software work products by assisting engineers to recognize and fix their own defects early in the software development process.

Types of defects that are easier to find during static testing are: deviations from standards, missing requirements, design defects, non-maintainable code and inconsistent interface specifications. Note that in contrast to dynamic testing, static testing finds defects rather than failures.

Benefits of Static Testing

Using static testing techniques to find defects and then fixing those defects promptly is almost always much cheaper for the organization than using dynamic testing to find defects in the test object and then fixing them, especially when considering the additional costs associated with updating other work products and performing confirmation and regression testing.

- Detecting and correcting defects more efficiently, and prior to dynamic test execution
- Identifying defects which are not easily found by dynamic testing
- Preventing defects in design or coding by uncovering inconsistencies, ambiguities, contradictions, omissions, inaccuracies, and redundancies in requirements
- Increasing development productivity (e.g., due to improved design, more maintainable code)
- Reducing development cost and time
- Reducing testing cost and time
- Reducing total cost of quality over the software's lifetime, due to fewer failures later in the lifecycle or after delivery into operation
- Improving communication between team members in the course of participating in reviews

Static Testing Types

Static testing - testing a work product without the work product code being executed.

Static analysis - the process of evaluating a component or system without executing it, based on its form, structure, content, or documentation.

Review – a type of static testing in which a work product or process is evaluated by one or more individuals to detect defects or to provide improvements

Informal review – a type of review that does not follow a defined process and has no formally documented output (as a ad hoc review).

Formal review – a type of review that follows a defined process with a formally documented output.



Formal Review Types

walkthrough

A type of review in which an author leads members of the review through a work product and the members ask questions and make comments about possible issues.

inspection

A type of formal review to identify issues in a work product, which provides measurement to improve the review process and the software development process.

technical review

A formal review by technical experts that examine the quality of a work product and identify discrepancies from specifications and standards.

peer review

A review performed by others with the same abilities to create the work product



User and Business Requirements

User Requirements

AS <PERSON>,

I WANT <WHAT?>,

SO THAT <VALUE>

User Requirements

REMOVE A CLASS FROM A TRAINER'S SCHEDULE / 11 HOURS

AS A TRAINER,

I WANT TO USE THE MOBILE APP TO REMOVE A CLASS THAT I TEACH FROM MY SCHEDULE

SO THAT MY STUDENTS KNOW THEY SHOULDN'T SHOW UP AT THAT TIME

| TASK | 3 HOURS |
|--|---------|
| MODIFY THE DATABASE LIBRARY AND DATABASE TO ALLOW TRAINER CLASSES TO BE REMOVED AND LOG THE ACTION | |

| TASK | 6 HOURS |
|---|---------|
| ADD A "REMOVE CLASS" API CALL TO THE TRAINER SCHEDULE SERVICE | |

| TASK | 2 HOURS |
|--|---------|
| UPDATE THE MOBILE APP UI TO ADD A "REMOVE CLASS" BUTTON TO THE PAGE THAT DISPLAYS A CLASS, HAVE IT CALL THE NEW API CALL TO REMOVE THE CLASS | |

User Requirements

Loyalty system 2022

As a registered user of Udemy
I want to use Promo Code
So that I will get the discount

Acceptance Criteria #1

GIVEN As a user how successfully completed courses
WHEN I want to receive a Promo Code
AND I will use it in the near future to pay for the new courses

| Title: | Priority: | Estimate: |
|---|-----------|-----------|
| User Story: As a [description of user], I want [functionality] so that [benefit]. | | |
| Acceptance Criteria: Given [how things begin] When [action taken] Then [outcome of taking action] | | |

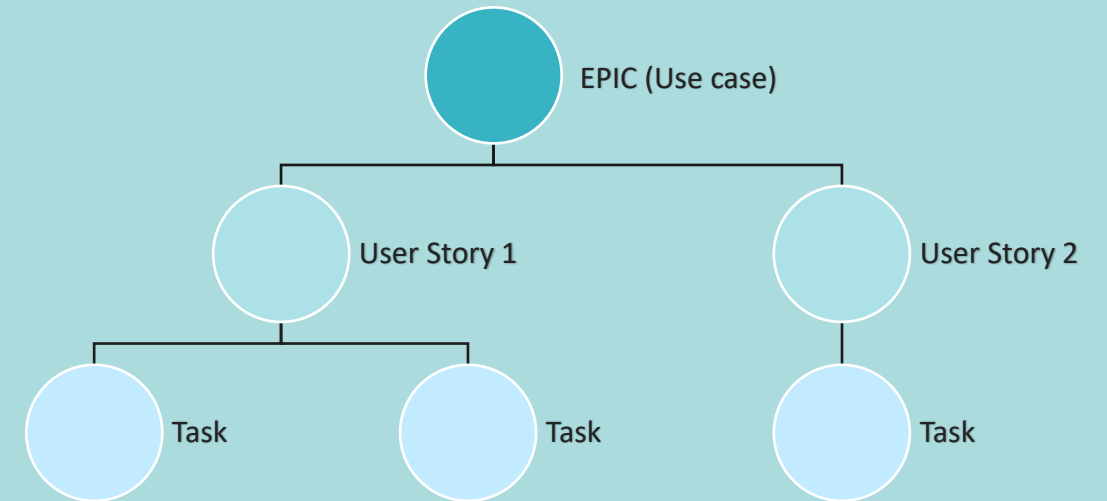
User Requirements

EPIC

A large user story that cannot be delivered as defined within a single iteration or is large enough that it can be split into smaller user stories.

USER STORY

A user or business requirement consisting of one sentence expressed in the everyday or business language which is capturing the functionality a user needs, the reason behind it, any non-functional criteria, and also including acceptance criteria.



INVEST

The INVEST mnemonic for Agile software development projects was created by Bill Wake as a reminder of the characteristics of a good quality Product Backlog Item (commonly written in user story format, but not required to be) or PBI for short. Such PBIs may be used in a Scrum backlog, Kanban board or XP project.

Independent

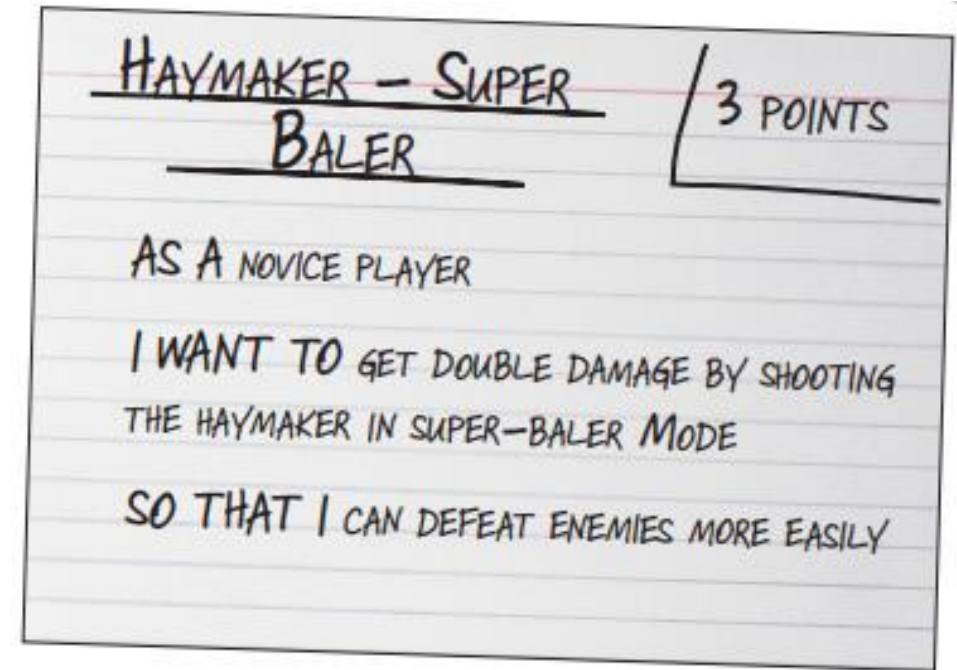
Negotiable

Valuable

Estimable

Small

Testable



Traceability

The degree to which a relationship can be established between two or more work products

| Req No | Req Desc | Testcase ID | Status |
|--------|--------------------------|--|---|
| 123 | Login to the application | TC01,TC02,TC03 | TC01-Pass TC02-Pass |
| 345 | Ticket Creation | TC04,TC05,TC06,TC07,TC08,TC09 TC010 | TC04-Pass TC05-Pass TC06-Pass TC06-Fail TC07-No Run |
| 456 | Search Ticket | TC011,TC012,TC013,TC014 | TC011-Pass TC012-Fail TC013-Pass TC014-No Run |

Estimation

an activity to calculate and approximate time, resources and expenses needed to complete test execution in a specified environment

Team One

Scope estimation

0

1/2

1

2

3

5

8

13

20

40

100

?

C

Waiting for moderator to finalise vote

Players: 00:02:04

Artem 00:01:58 13

yura 00:01:39 13

Max 00:01:13 8

Oleksii Novakov 00:01:40 8

Vladyslav Aleks... 00:01:14 8

Oleksandr Koste... 00:01:30 8

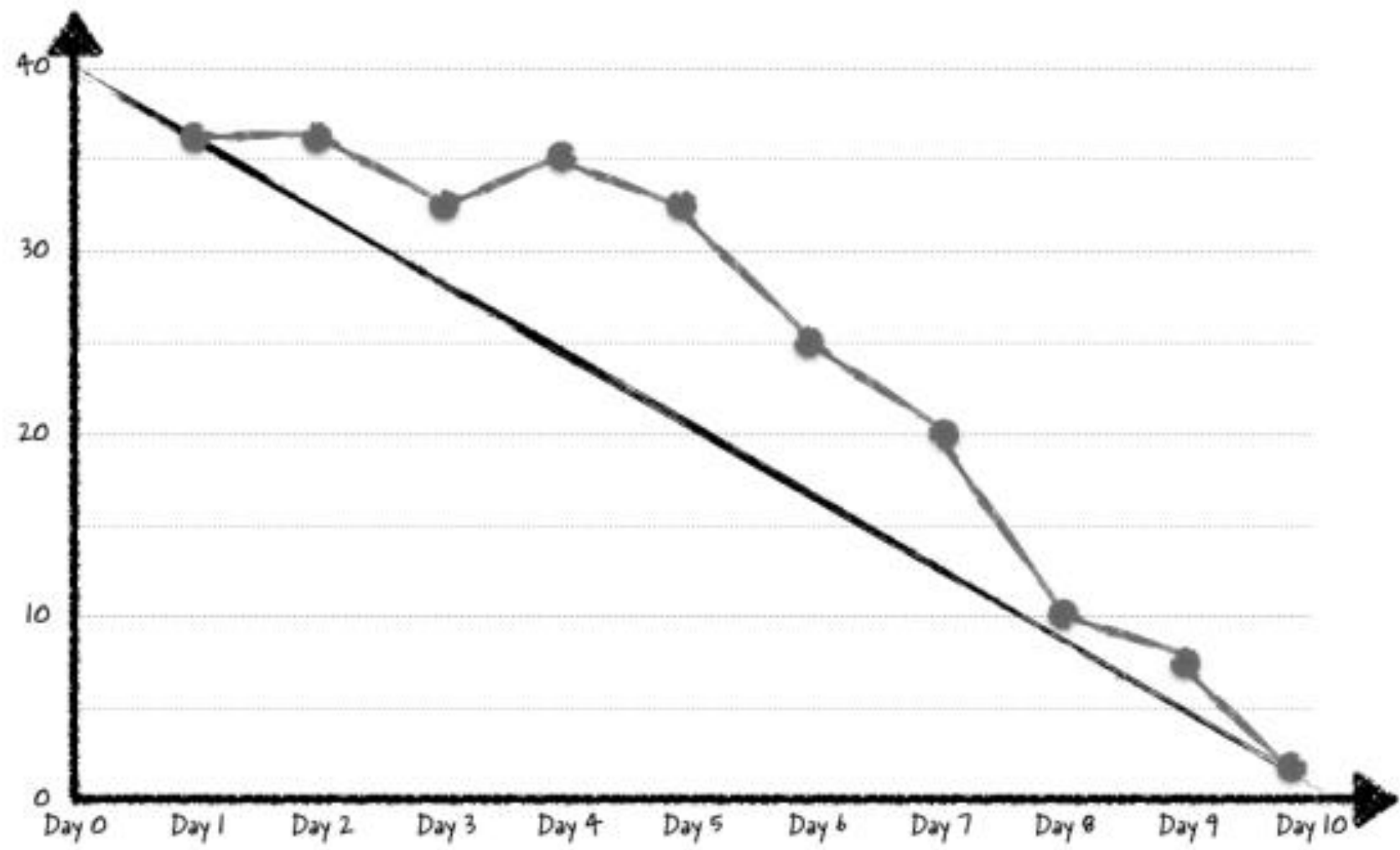
mchervonik 00:01:17 8

Invite a teammate

Active Stories 1

Completed Stories 0

All Stories 0



Home Task

to read:

1. This presentation
2. Software Testing (Куліков) 29 – 63 pp
3. Syllabus 45 – 54 pp (to exclude 3.2.1 and 3.2.2)

to learn:

1. Vocabulary

to do:

1. Quiz

Questions

Thank you