**Branch :- Computer Sci. & Engg. Class :- III Year**

**Subject :- Software Engineering Lab Sem :- VI**

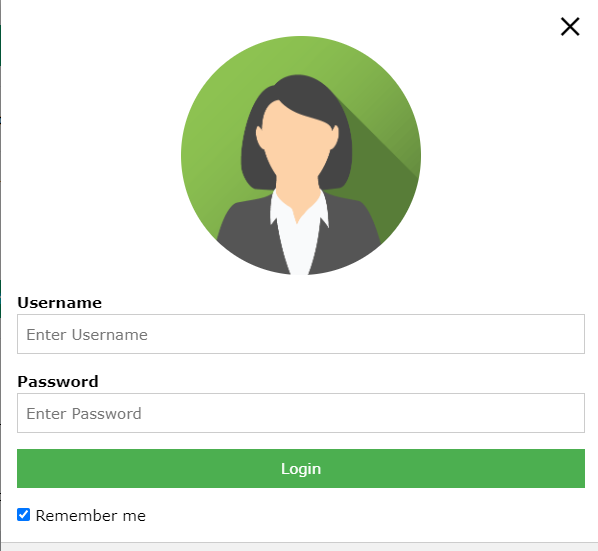
**Teacher Manual**

**PRACTICAL NO 9**

**Aim:** Write Down the Test Case scenario for Login page authentication.

**Theory:**

**Case Study**



Let us see how we can design test cases to verify the "User Login" feature. The simplest scenario is when both user name and password have been typed in correctly. The outcome will be that the user could then avail all features of LIS. However, there could be multiple unsuccessful conditions:

* User ID is wrong
* Password is wrong
* User ID & password are wrong
* Wrong password given twice consecutively
* Wrong password given thrice consecutively
* Wrong password given thrice consecutively, and security question answered correctly
* Wrong password given thrice consecutively, and security question answered incorrectly

We would create test case for the above stated login scenarios. These test cases together would constitute a test suite to verify the concerned requirement. Table 1 shows the details of this test suite.

| Table 1: A test suite to verify the "User Login" feature | | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| **#** | | **TS1** | | | | |
| **Title** | | **Verify "User Login" functionality** | | | | |
| **Description** | | **To test the different scenarios that might arise while an user is trying to login** | | | | |
|  | | | | | | |
| **#** | **Summary** | **Dependency** | **Pre-condition** | **Post-condition** | **Execution Steps** | **Expected Output** |
| TC1 | Verify that user already registered with the LIS is able to login with correct user ID and password |  | Employee ID *149405* is a registered user of LIS; user's password is *this\_is\_password* | User is logged in | 1. Type in employee ID as *149405* 2. Type in password *this\_is\_password* 3. Click on the 'Login' button | "Home" page for the user is displayed |
| TC2 | Verify that an unregistered user of LIS is unable to login |  | Employee ID *149405xx* is not a registered user of LIS | User is not logged in | 1. Type in employee ID as *149405xx* 2. Type in password *whatever* 3. Click on the 'Login' button | The "Login" dialog is shown with a *"Login failed! Check your user ID and password"* message |
| TC3 | Verify that user already registered with the LIS is unable to login with incorrect password |  | Employee ID *149405* is a registered user of LIS; user's password is *this\_is\_password* | User is not logged in | 1. Type in employee ID as *149405* 2. Type in password *whatever* 3. Click on the 'Login' button | The "Login" dialog is shown with a *"Login failed! Check your user ID and password"* message |
| TC4 | Verify that user already registered with the LIS is unable to login with incorrect password give  n twice consecutively | TC3 | This test case is executed after execution of TC3 before executing any other test case | User is not logged in | 1. Type in employee ID as *149405* 2. Type in password *whatever2* 3. Click on the 'Login' button | The "Login" dialog is shown with a *"Login failed! Check your user ID and password"* message |
| TC5 | Verify that user already registered with the LIS is unable to login with incorrect password given thrice consecutively | TC4 | This test case is executed after execution of TC4 before executing any other test case | User is not logged in | 1. Type in employee ID as *149405* 2. Type in password *whatever3* 3. Click on the 'Login' button | The "Login" dialog is shown with a *"Login failed! Check your user ID and password"* message; the security question and input box for the answer are displayed |
| TC6 | Verify that a registered user can login after three consecutive failures by correctly answering the security question | TC5 | This test case is executed after execution of TC6 before executing any other test case. Answer to the security question is *my\_answer*. | Email sent containing new password. The email is expected to be received within 2 minute. | 1. Type in the answer as *my\_answer* 2. Click on the 'Email Password' button | Login dialog is displayed; an email containing the new password is received |
| TC7 | Verify that a registered user's account is blocked after three consecutive failures and answering the security question incorrectly |  | Execute the test cases TC3, TC4, and TC5 once again (in order) before executing this test case | User account has been blocked | 1. Type in the answer as *not\_my\_answer* 2. Click on the 'Email Password' button | The message *"Your account has been blocked! Please contact the administrator."* appears |

In a similar way, test suites corresponding to other user requirements could be created as well. A good test plan can reduce the burden of testing team by specifying what exactly they should focus on.

**Conclusion:** Thus we have studied designing a test cases and test suites.

**Branch :- Computer Sci. & Engg. Class :- III Year**

**Subject :- Software Engineering Lab Sem :- VI**

**Teacher Manual**

**PRACTICAL NO 10**

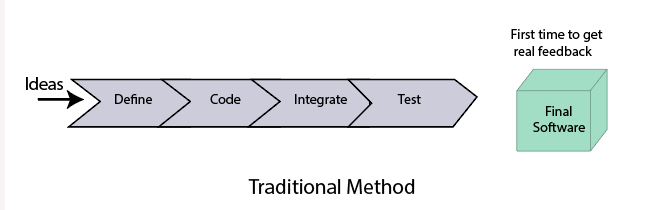
**Aim :** Introduction to Agile Methodology

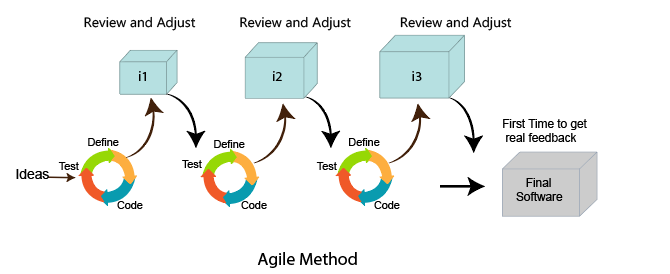
**Theory :**

An agile methodology is an iterative approach to software development. Each iteration of agile methodology takes a short time interval of 1 to 4 weeks. The agile development process is aligned to deliver the changing business requirement. It distributes the software with faster and fewer changes.

The single-phase software development takes 6 to 18 months. In single-phase development, all the requirement gathering and risks management factors are predicted initially.

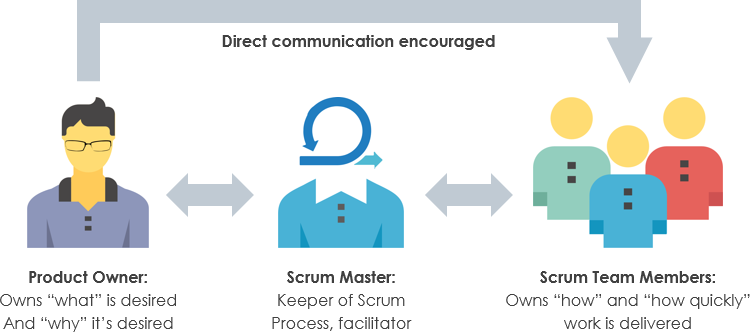
The agile software development process frequently takes the feedback of workable product. The workable product is delivered within 1 to 4 weeks of iteration.





A [**Scrum Team**](https://www.visual-paradigm.com/scrum/what-is-scrum-team/) is a collection of individuals working together to deliver the requested and committed product increments which consists of three [**roles**](https://www.visual-paradigm.com/scrum/what-are-the-three-scrum-roles/):

* [**Product Owner**](https://www.visual-paradigm.com/scrum/what-is-project-owner-role-in-scrum/)
* [**Scrum Master**](https://www.visual-paradigm.com/scrum/what-is-scrum-master/)
* Development Team



Self-Organizing and Cross-Functional

The [**Scrum**](https://www.visual-paradigm.com/scrum/scrum-in-3-minutes/) Teams are **self-organizing** and **cross-functional**:

* **Self-organizing teams** choose how best to accomplish their work, rather than being directed by others outside the team.
* **Cross-functional teams** have all competencies needed to accomplish the work without depending on others not part of the team. The team model in Scrum is designed to optimize flexibility, creativity, and productivity.

**How Scrum Team Can Work Together Effectively?**

Scrum Teams deliver products iteratively and incrementally, maximizing opportunities for feedback. Incremental deliveries of “Done” product ensure a potentially useful version of working product is always available.

Scrum is defined by a group of principles (or 5 scrum values) that should be understood as simple guidelines for working together more effectively as a team. They are:

* **Courage**—especially when it comes to solving hard problems
* **Focus**
* **Commitment**to the shared team goals
* **Respect** for your team members
* **Openness** about work and any challenges that might come up.

**Scrum Key Concepts**

Scrum consist a series of [**sprint**](https://www.visual-paradigm.com/scrum/what-is-sprint-in-scrum/) or development iteration. Each scrum sprint involves the same process: Role (people), Events (meetings) and [**Artifacts**](https://www.visual-paradigm.com/scrum/what-are-scrum-artifacts/) (tangible by-products).

* Sprints are development cycles that repeat until your project is complete.
* Requirements (often in the form of user stories) are developed, tested, integrated, and approved within each sprint. And the process continues sprint after sprint.

As mentioned before, the Scrum framework is consists of three distinct categories**: roles**, **events**, and **artifacts**:

Scrum Roles

The Scrum framework is defined by three core roles: the Development Team, the Scrum Master, and the Product Owner.

The Product Owner

The Product Owner is responsible for maximizing the value of the product and the work of the Development Team. How this is done may vary widely across organizations, Scrum Teams, and individuals.

Role of Product Owner

The Product Owner is the sole person responsible for managing the [**Product Backlog**](https://www.visual-paradigm.com/scrum/what-is-product-backlog-in-scrum/). Product Backlog management includes:

* Clearly expressing Product Backlog items;
* Ordering the items in the Product Backlog to best achieve goals and missions;
* Optimizing the value of the work the Development Team performs;
* Ensuring that the Product Backlog is visible, transparent, and clear to all, and shows what the Scrum Team will work on next; and,
* Ensuring the Development Team understands items in the Product Backlog to the level needed.

**The Scrum Master**

The Scrum Master is responsible for ensuring Scrum is understood and enacted. Scrum Masters do this by ensuring that the Scrum Team adheres to Scrum theory, practices, and rules.

The Scrum Master is a servant-leader for the Scrum Team. The Scrum Master helps those outside the Scrum Team understand which of their interactions with the Scrum Team are helpful and which aren’t. The Scrum Master helps everyone change these interactions to maximize the value created by the Scrum Team.

Role of Scrum Master

The Scrum Master serves the Product Owner in several ways, including:

* Finding techniques for effective Product Backlog management;
* Helping the Scrum Team understand the need for clear and concise Product Backlog items;
* Understanding product planning in an empirical environment;
* Ensuring the Product Owner knows how to arrange the Product Backlog to maximize value;
* Understanding and practicing agility; and,
* Facilitating [**Scrum events**](https://www.visual-paradigm.com/scrum/what-are-scrum-events/) as requested or needed.

Scrum Master Service to the Development Team

The Scrum Master serves the Development Team in several ways, including:

* Coaching the Development Team in self-organization and cross-functionality;
* Helping the Development Team to create high-value products;
* Removing impediments to the Development Team’s progress;
* Facilitating Scrum events as requested or needed; and,
* Coaching the Development Team in organizational environments in which Scrum is not yet fully adopted and understood.

Scrum Master Service to the Organization

* The Scrum Master serves the organization in several ways, including:
* Leading and coaching the organization in its Scrum adoption;
* Planning Scrum implementations within the organization;
* Helping employees and stakeholders understand and enact Scrum and empirical product development;
* Causing change that increases the productivity of the Scrum Team; and,
* Working with other Scrum Masters to increase the effectiveness of the application of Scrum in the organization.

The Development Team

The Development Team consists of professionals who do the work of delivering a potentially releasable Increment of “Done” product at the end of each Sprint. Only members of the Development Team create the Increment.

Development Teams are structured and empowered by the organization to organize and manage their own work. The resulting synergy optimizes the Development Team’s overall efficiency and effectiveness.

Role of Development Team

Development Teams have the following characteristics:

* They are self-organizing. No one (not even the Scrum Master) tells the Development Team how to turn Product Backlog into Increments of potentially releasable functionality;
* Development Teams are cross-functional, with all of the skills as a team necessary to create a product Increment;
* Scrum recognizes no titles for Development Team members other than Developer, regardless of the work being performed by the person; there are no exceptions to this rule;
* Scrum recognizes no sub-teams in the Development Team, regardless of particular domains that need to be addressed like testing or business analysis; there are no exceptions to this rule; and,
* Individual Development Team members may have specialized skills and areas of focus, but accountability belongs to the Development Team as a whole.

Scrum Events

The Scrum framework is marked by five Events. These are the Sprint, [**Sprint Planning**](https://www.visual-paradigm.com/scrum/what-is-sprint-planning/), [**Daily Scrum**](https://www.visual-paradigm.com/scrum/daily-scrum-meeting-quick-guide/), [**Sprint Review**](https://www.visual-paradigm.com/scrum/what-is-sprint-review/), and [**Sprint Retrospective**](https://www.visual-paradigm.com/scrum/what-is-sprint-retrospective-meeting/).

* A **sprint** (also called an iteration) is the basic unit of development in Scrum. The sprint is a [**time-boxed**](https://www.visual-paradigm.com/scrum/what-is-time-boxing-in-scrum/) effort; that is, it is restricted to a specific duration. The duration is fixed in advance for each sprint and is normally between one week and one month, with two weeks being the most common.
* **Sprint planning** is an event in the Scrum framework where the team determines the product backlog items they will work on during that sprint and discusses their initial plan for completing those product backlog items.
* A **daily stand-up**(also called daily Scrum meeting) is a short organizational meeting that is held each day. The meeting, generally limited to between five and fifteen minutes long, is sometimes referred to as a stand-up, a morning roll-call or a daily scrum.
* The **Sprint Review** takes place after a Sprint ends. During Review, the Product Owner explains what planned work either was or was not completed during the Sprint. The team then presents completed work and talks through what went well and how problems were solved.
* The **Sprint Retrospective** is held after the sprint review at the end of each sprint. It offers the team an opportunity to inspect itself and create a plan for improvements to be enacted during the next Sprint.

Scrum Artifacts

Artifacts are just physical records that provide project details. Scrum Artifacts include the Product Backlog, [**Sprint Backlog**](https://www.visual-paradigm.com/scrum/what-is-sprint-backlog-in-scrum/), and Product Increments.

* The **Product Backlog** is a prioritized list of features, defects, or technical work that has yet to be worked on. It should be work that is considered valuable from the product owner’s perspective.
* The **Product Owner** and the rest of the team work together to review the Product Backlog and make adjustments when necessary, as product requirements change and evolve.
* The **Sprint Backlog** is a list of all items from the Product Backlog to be worked on during a Sprint. This list is put together by prioritizing items from the Product Backlog until the team feels they’ve reached their capacity for the Sprint. Team members sign up for tasks in the Sprint Backlog based on skills and priorities, following the self-organizing Scrum framework.
* A Product **Increment**is the sum of product work completed during a Sprint, combined with all work completed during previous Sprints. The goal of a Sprint is to produce a Done Product Increment. It’s up to the Scrum team to agree on what defines an Increment’s “Done” status, but all team members need to agree on and understand the definition.

**Scrum Lifecycle**

The Scrum Lifecycle starts with a prioritized backlog, but does not provide any guidance as to how that backlog is developed or prioritized, which consists of a series of Sprints that yield the end result as a potentially shippable product increment as shown in the Figure below.

Inside of these sprints, all of the activities necessary for the development of the product occur on a small subset of the overall product.  Below is a description of the key steps in the Scrum Lifecycle:

The agile scrum framework

* The business sets the priorities and the team organizes themselves to determine the best way to deliver the highest priority features.
* The Product Owner represents the Business side and is responsible for the maintaining the list of product features wish list called a product backlog and sets priorities for development.
* During sprint planning, the team pulls a small chunk from the top of that wish list, a sprint backlog, and decides how to implement those pieces.
* The Scrum process is based on iterative cycles called Sprints which typically last 2-4 weeks during which the product is designed, coded and tested, while meeting every day to assess its progress (daily Scrum).
* Along the way, the Scrum Master keeps the team focused on its goal.
* At the end of the sprint, the work should be potentially shippable for showing to a stakeholder.
* The sprint ends with a sprint review and retrospective meetings.
* As the next sprint begins, the team chooses another chunk of the product backlog and begins working again.
* The project will either goes on until it is completed the entire wish list, or it will be stopped due to the time or budget has already been exhausted.
* This goes on until the project is deemed complete, either by stopping to work (deadline, budget, etc.) or by completing the entire wish list.

**Conclusion**: Thus we have studied the concept of Agile Methodology