BAHRIA UNIVERSITY, Karachi Campus)

**Department of Software Engineering**

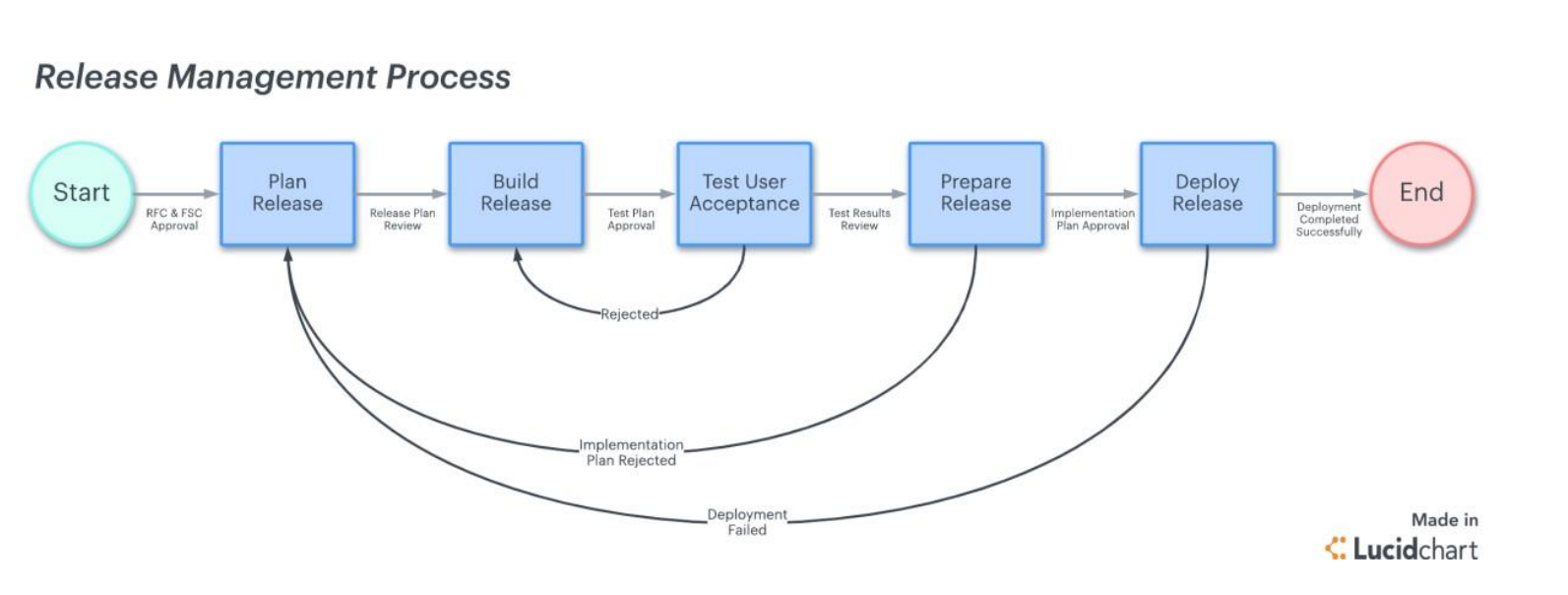
**ASSIGNMENT # 0**3 **– Fall 2022**

|  |  |
| --- | --- |
| Course Title: **Software Construction** | Course Code: **SEC-311** |
| Class: **BSE – 5(B)** | Shift: **Morning** |
| Course Instructor: **Dr. Salahuddin Shaikh** | Date: **14th Dec 2023** |
| Due Date: **28st December 2023** | Max. Marks: **5 Marks** |

|  |  |
| --- | --- |
| Name: **Muhammad Shoaib Akhter Qadri** | Enrollment No:**02-131212-009** |

**Question No. 1: [CLO#03, 5 marks]**

Considering the following diagram, evaluate release management process according to your opinion.



Solution:

## 1. Plan release

The planning stage may be the most time intensive as this is where our entire release is structured from start to finish. A robust release plan will help our team stay on track and ensure our meet standards and requirements.

We can approach a release plan. One of the most popular release management methods is the systems development life cycle (SDLC).

The SDLC helps software developers plan, develop, maintain, and replace software systems with a high degree of efficiency and quality. We can use the SDLC can be used in conjunction with or in place of other project management processes.

The workflow can explain at a glance how the whole release is staged and how each team member plays a part. Our release plan should include:

* Timelines
* Delivery dates
* Requirements
* The overall scope of the project

Once our plan is sketched out, present it to all relevant stakeholders (our team, product manager, and high-level leaders) for review. Get their feedback on any gaps or problems they see in the requirements or scope.

Once the plan is approved and finalized, we can put it into action.

## 2. Build release

With the release plan finalized, we can start designing and building the product for release. This is the actual “development” of the product based on the requirements outlined in the release plan.

Once we have a working version of our product, it’s time to subject the build to real-world scenario testing.

As the team builds out the product, it is sent (usually automatically) to a testing environment for user acceptance, or stage three in the release management process. This allows the team to identify any bugs or issues that may arise in a real-world environment.

As users identify issues, the team sends the build back for development at stage two. In other words, within the iterative release management process, the work may flow from stage two to stage three and back again until the release is approved.

## 3. User acceptance testing

User acceptance testing (UAT), is the point when the end users get to actually use the product and give feedback. This is often done as a free beta trial online or shared with a larger group of employees within the company.

User acceptance testing is the most crucial step to release management because of the amount of data collected and fixes required in order to get the build to where it needs to be for the official launch.

As noted earlier, this is part of an iterative process. As bugs are identified, the team goes back to the drawing board to fix the issues and redesign the build for greater integrity. The build must pass the UAT stage to be considered for final implementation and release.

## 4. Prepare release

This step is to put the finishing touches on the product, taking into account everything that was learned in UAT. Release preparation also includes a final review by the quality assurance (QA) team.

During the review, the QA team will conduct final checks to ensure the build meets the minimum acceptable standards and business requirements outlined in the release plan.

Although UAT and quality assurance can’t always replicate every scenario that might occur once the product is launched, these steps hopefully find the most common bugs so that your team can better anticipate and prevent any problems at launch.

Once the review is completed, the functional team will validate the findings and finalize the release for deployment. Before the build can deploy into a live environment, it must be approved by the product owner.

## 5. Deploy release

The big day has finally arrived and here is where all your team’s hard work pays off. It’s time to release our product into the wilds of the live production environment.

Besides simply sending the build out into production, the deployment stage also includes messaging and education on the product to both the end user and your company at large.

For instance, we notify users of changes with the release and how to operate within the new features. Depending on how significant the changes were, you may need to provide robust and ongoing training to get everyone up to speed.

This is especially important for internal releases where employees using the software need to understand it to do their work efficiently and productively.

Finally, during the deployment stage, the development team should meet to assess the release’s performance and discuss how the deployment went. If there are any lingering issues, identify and document them for the team to address in the next iteration.

Release management is a constantly changing process. Each deployment life cycle is an opportunity to refine everything from your workflow to your checklist as your team discovers what roadmap works best for what kind of launch—and what doesn’t.

The collaborative platform makes it easy for team members—from developers and product owners to executive stakeholders—to view the high-level plan and get at-a-glance insights into their progress so everyone is on the same page.

.