REFERAT Application Lifecycle Management

fiabilitatea si testarea aplicatiilor software

Cuprins

[Application Lifecycle Management 3](#_Toc135295990)

[1.1 Introducere 3](#_Toc135295991)

[1.2 Notiuni si termeni 3](#_Toc135295992)

[1.3 Procesul practic 4](#_Toc135295993)

[1.4 Principii generale 15](#_Toc135295994)

[1.5 Intrebari 16](#_Toc135295995)

# Application Lifecycle Management



## Introducere

Application Lifecycle Management (ALM) este un proces care se refera la planificarea, dezvoltarea, testarea, implementarea si mentenanta unei aplicatii software. Acest proces acopera intregul ciclu de viata al unei aplicatii, incepand de la conceperea ideii, trecand prin dezvoltare si testare, pana la lansarea si mentinerea acesteia in productie. ALM implica gestionarea proiectului, gestionarea cerintelor, gestionarea versiunilor, testarea, urmarirea defectelor si gestionarea configuratiei.

ALM se concentreaza pe imbunatatirea colaborarii intre echipele de dezvoltare si operatii si are ca scop cresterea eficientei procesului de dezvoltare a software-ului. Prin utilizarea ALM, organizatiile pot imbunatati calitatea aplicatiilor lor, reducand in acelasi timp costurile si timpii de dezvoltare.

Micro Focus Application Lifecycle Management (ALM) este un set de instrumente software dezvoltate si comercializate de Micro Focus pentru dezvoltarea si testarea aplicatiilor. Acesta include instrumente pentru gestionarea cerintelor, planificarea testelor si testarea functionala, testarea performantei (cand este utilizat cu Performance Center), managementul dezvoltatorilor (prin medii precum Collabnet, TeamForge, Microsoft Visual Studio) si managementul defectelor.

## Notiuni si termeni

Managementul cererilor (Request management): In aceasta etapa, cerintele sunt colectate, analizate si prioritizate. Aceasta faza este cruciala, deoarece definirea cerintelor corecte asigura dezvoltarea produsului potrivit.

Managementul cerintelor (Requirements management): procesul de definire, capturare, analiza, validare si urmarire a cerintelor de software.

Managementul testelor (Test management): Acesta include planificarea si executarea testelor pentru a valida ca software-ul functioneaza asa cum este specificat in cerinte. Testele sunt efectuate pe intreaga durata a ciclului de viata al aplicatiei pentru a detecta erorile cat mai devreme posibil.

Managementul defectelor (Defect management): Defectele sunt raportate, urmarite si gestionate in aceasta faza. Acest proces este important pentru a asigura ca defectele sunt remediate in timp util si in mod corespunzator.

Managementul versiunilor (Version control): Aceasta etapa implica gestionarea codului sursa al aplicatiei si a modificarilor acestuia. Ofera suport pentru colaborarea in echipe si asigura ca fiecare versiune a aplicatiei poate fi monitorizata, gestionata si, in caz de probleme, revertata la o versiune anterioara.

Managementul configurarii (Configuration management): procesul de gestionare a configuratiei software-ului, inclusiv gestionarea versiunilor si modificarilor, in timpul ciclului de viata al dezvoltarii software-ului.

Managementul lansarii (Release management): procesul de pregatire si lansare a versiunilor de software in mediul de productie.

Managementul proiectelor (Project management): Aceasta etapa implica gestionarea intregului ciclu de viata al aplicatiei, asigurandu-se ca resursele sunt alocate in mod corespunzator, termenele sunt respectate si proiectul este livrat cu succes.

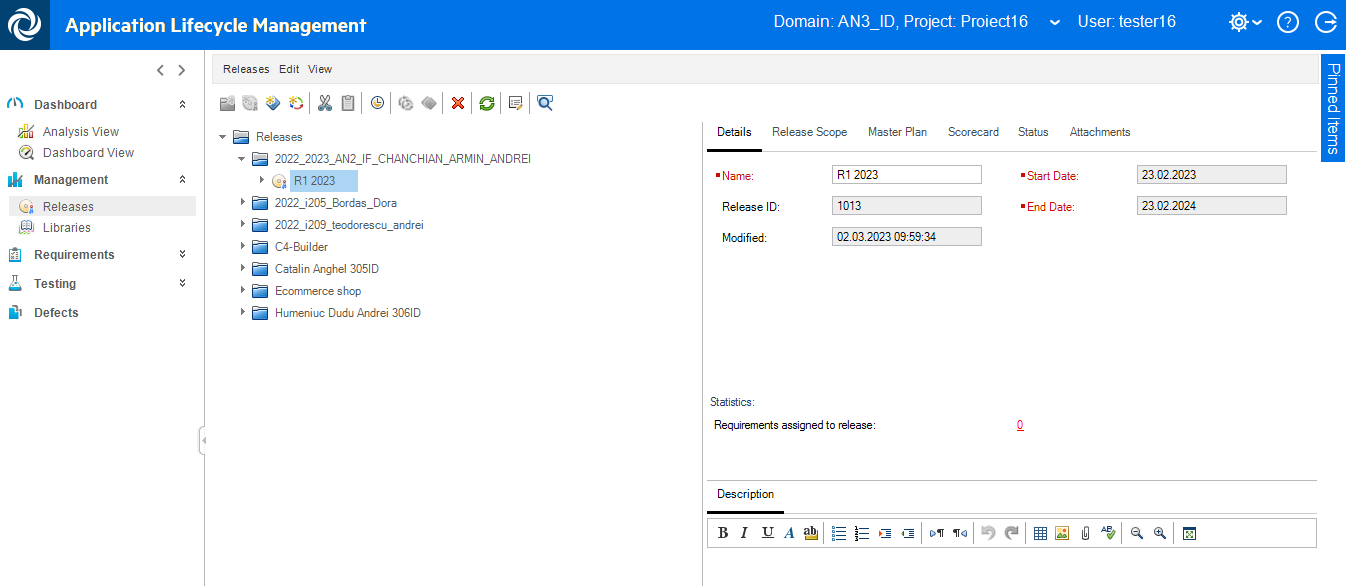
## Procesul practic

ALM poate fi rulat printr-un client specializat in acest sens sau direct din browser.

**Application Lifecycle Management rulat prin browser**

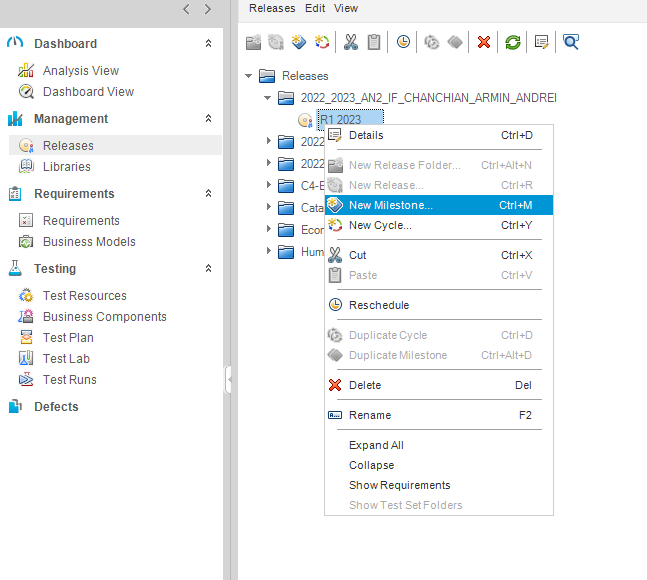
Mediul Micro Focus ALM este accesibil in urma conectarii la adresa de internet a serverului, urmat de autentificarea in urma introducerii datelor de utilizator, selectarea domeniului si a proiectului.

1. **Creare release user**

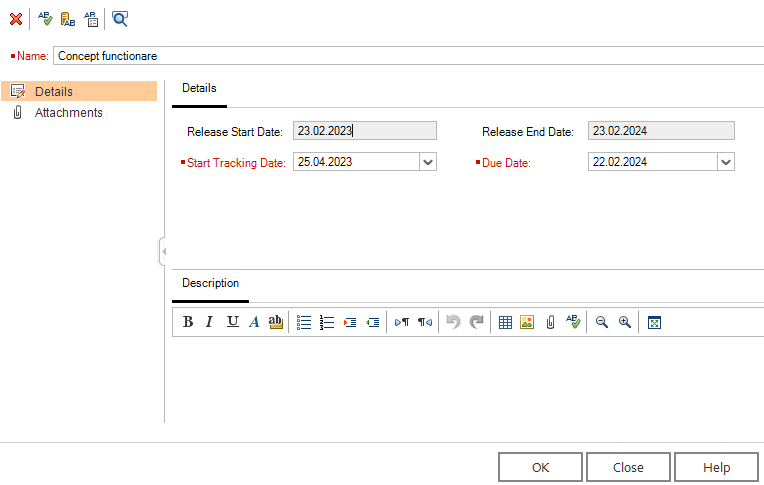
In categoria **Management** > **Releases** a fost constituit un nou director aferent utilizatorului iar mai apoi anexat acestui director, a fost creat un **Release**.

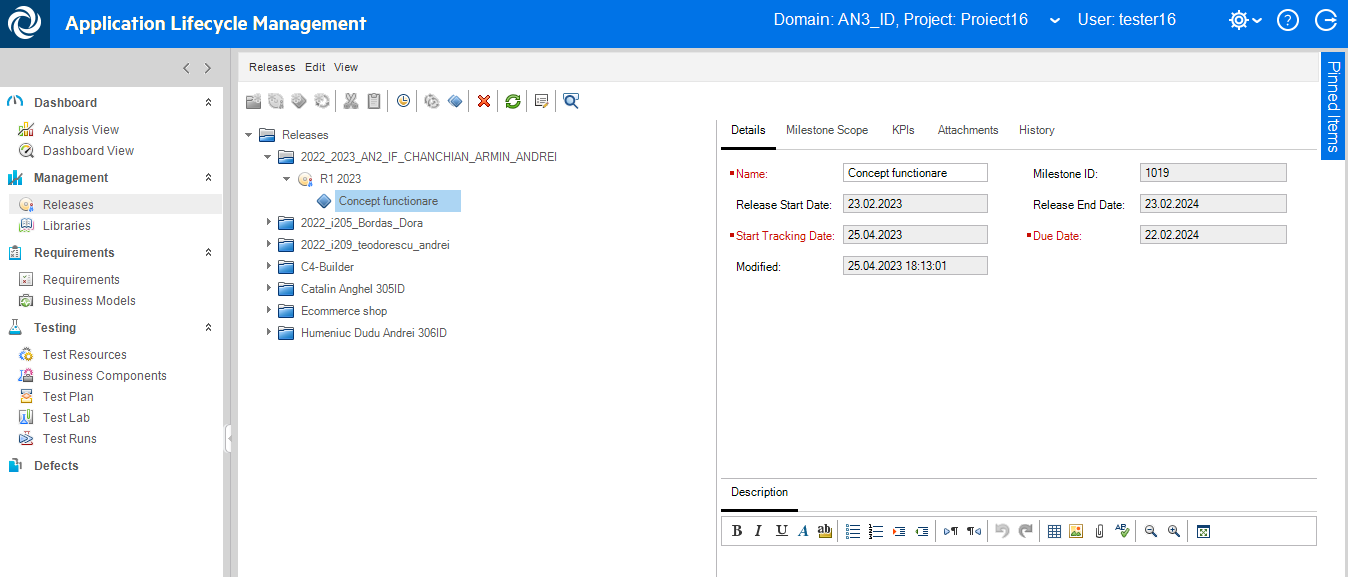
1. **Creare Milestone**

Release-ul are atasat un **Milestone**, ce reprezinta un obiectiv care are un termen limita.

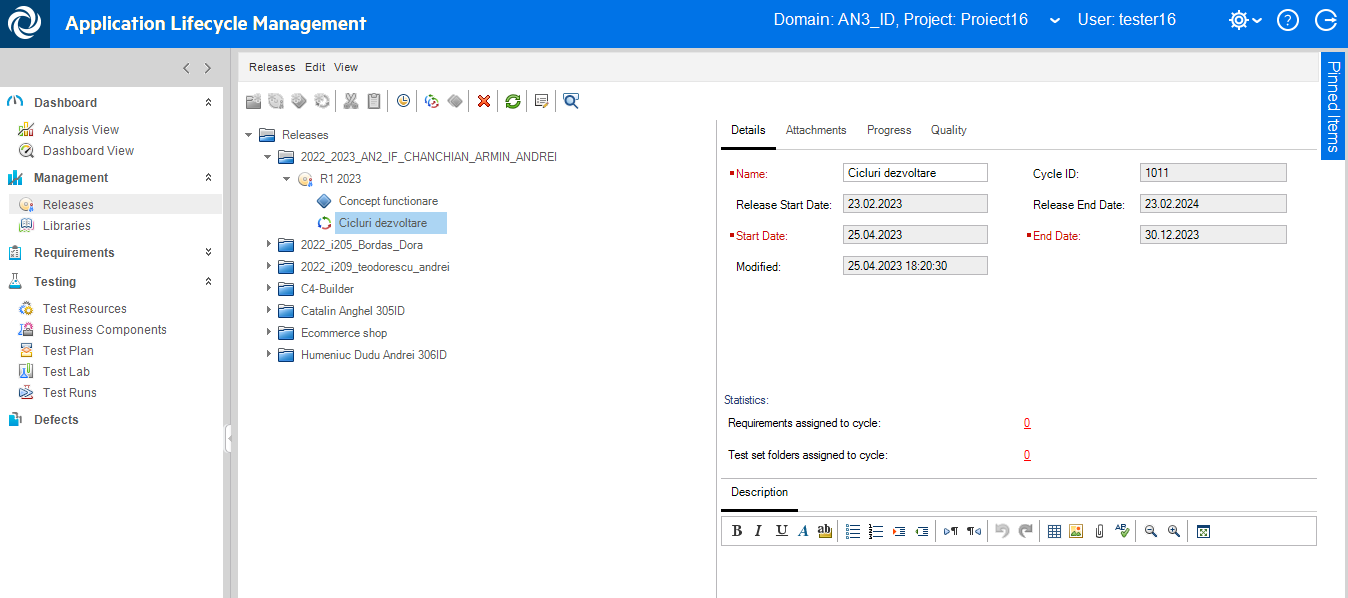


Release-ul este selectat, se apasa **New Milestone** si se completeaza.



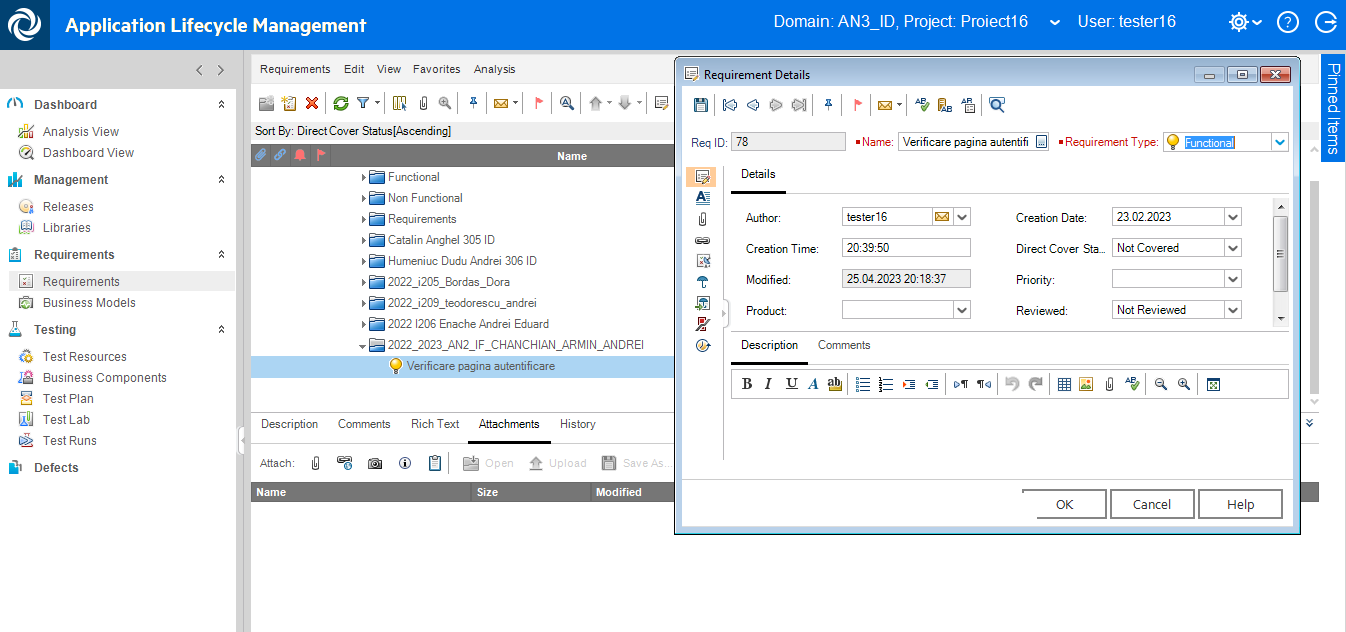


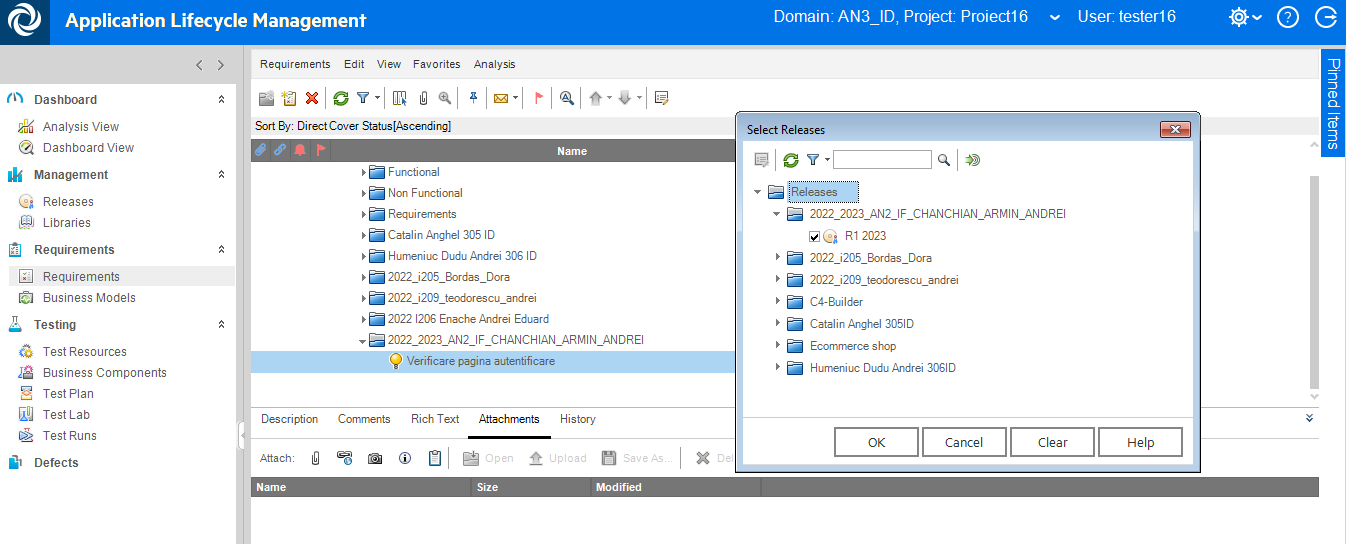
**3. Creare Release Cycle**

Acestui Release ii va fi adaugat un **Release Cycle**, ce reprezinta o metoda de planificare, urmarire si gestionare a lansarilor de software intr-un ciclu iterativ. Este adesea folosit pentru a defini durata ciclului de livrare, perioada de testare si stabilizare a software-ului, precum si procesele de aprobare si de livrare finala.

**4. Adaugare Requirements**

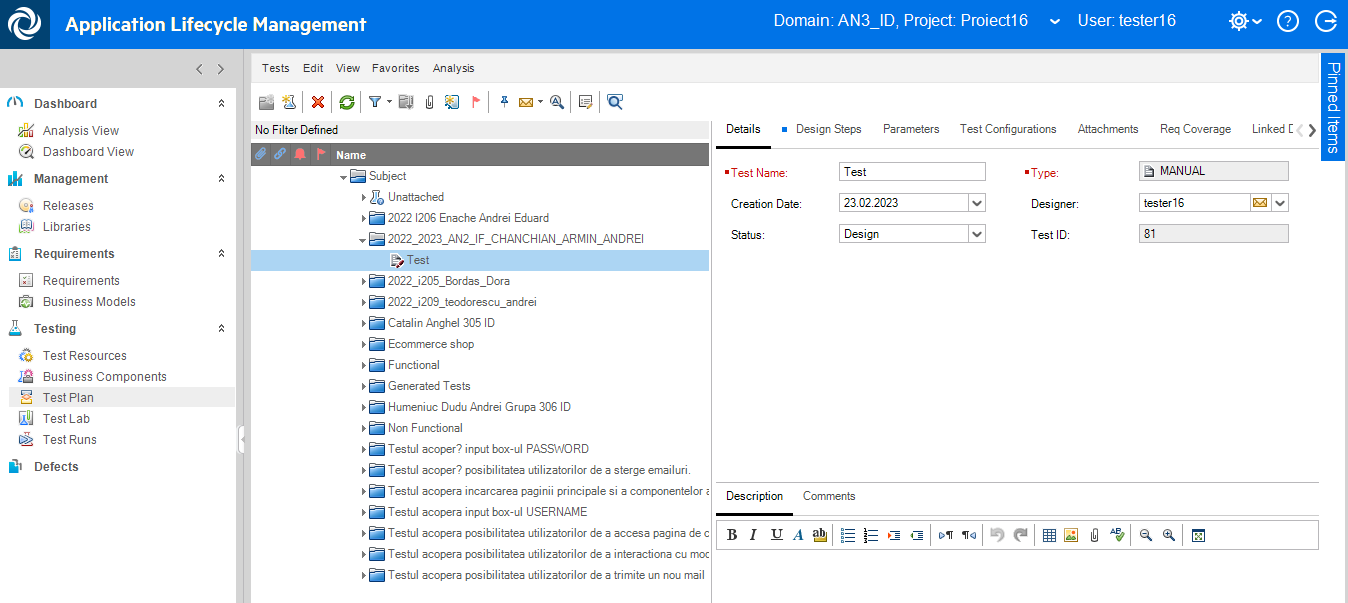
Release Cycle-urile anterior adaugate, este necesar sa li se anexeze niste **Cerinte** (**Requirements**). Acestea reprezinta specificatiile functionale sau non-functionale ale produsului software, care trebuie sa fie indeplinite pentru a satisface nevoile si asteptarile clientilor sau utilizatorilor finali.



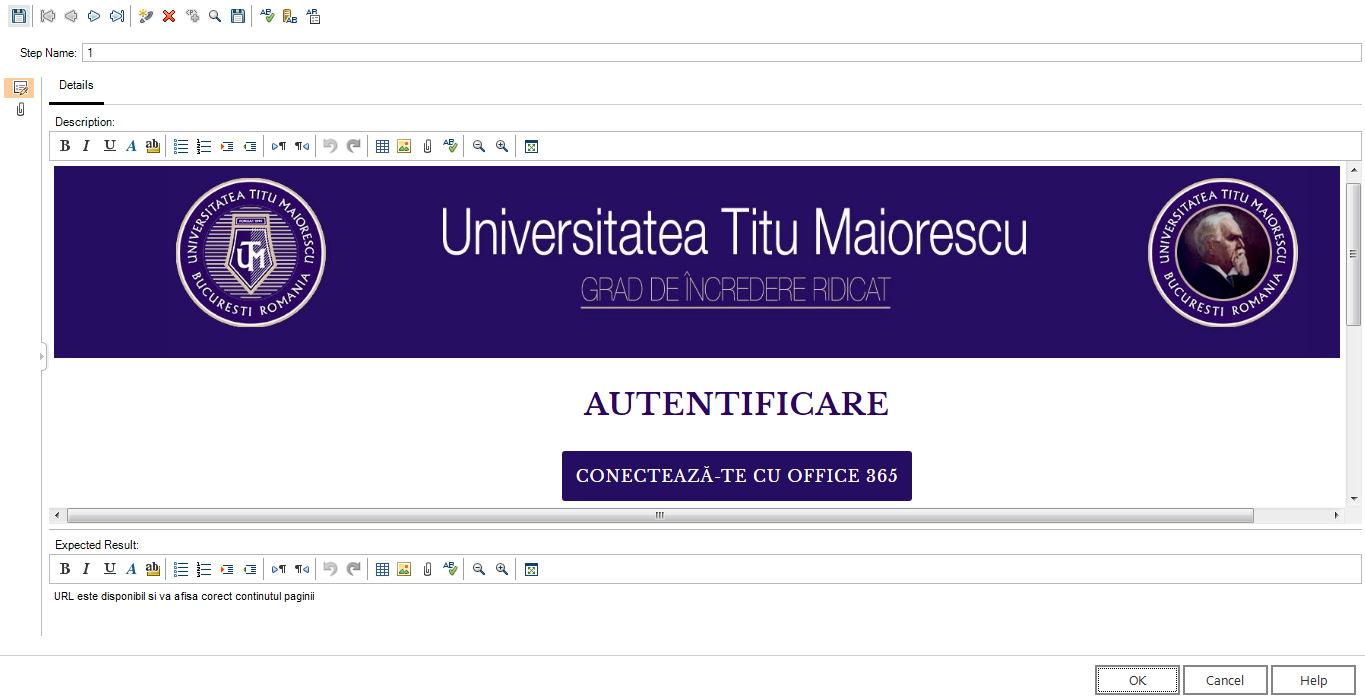
Requirements-urile se pot anexa Release-ului dorit, prin functia **Assign to Release**.

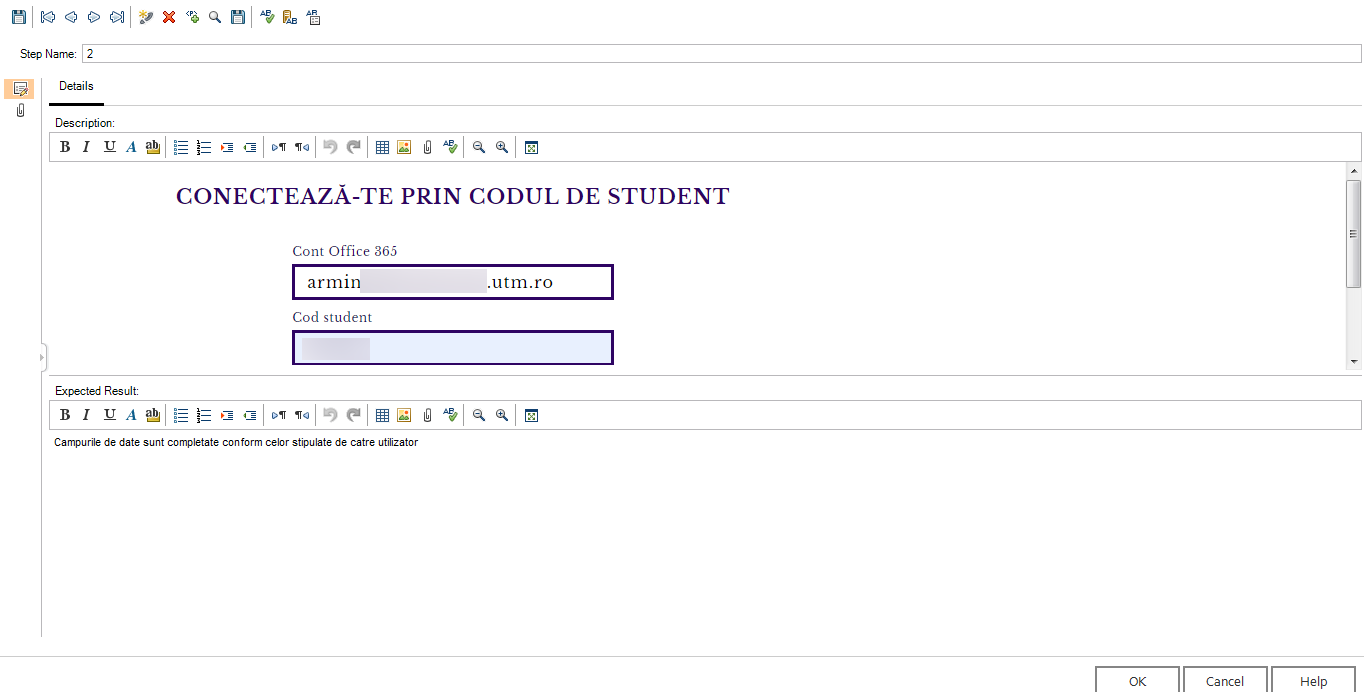
**6. Creare Test Plan**

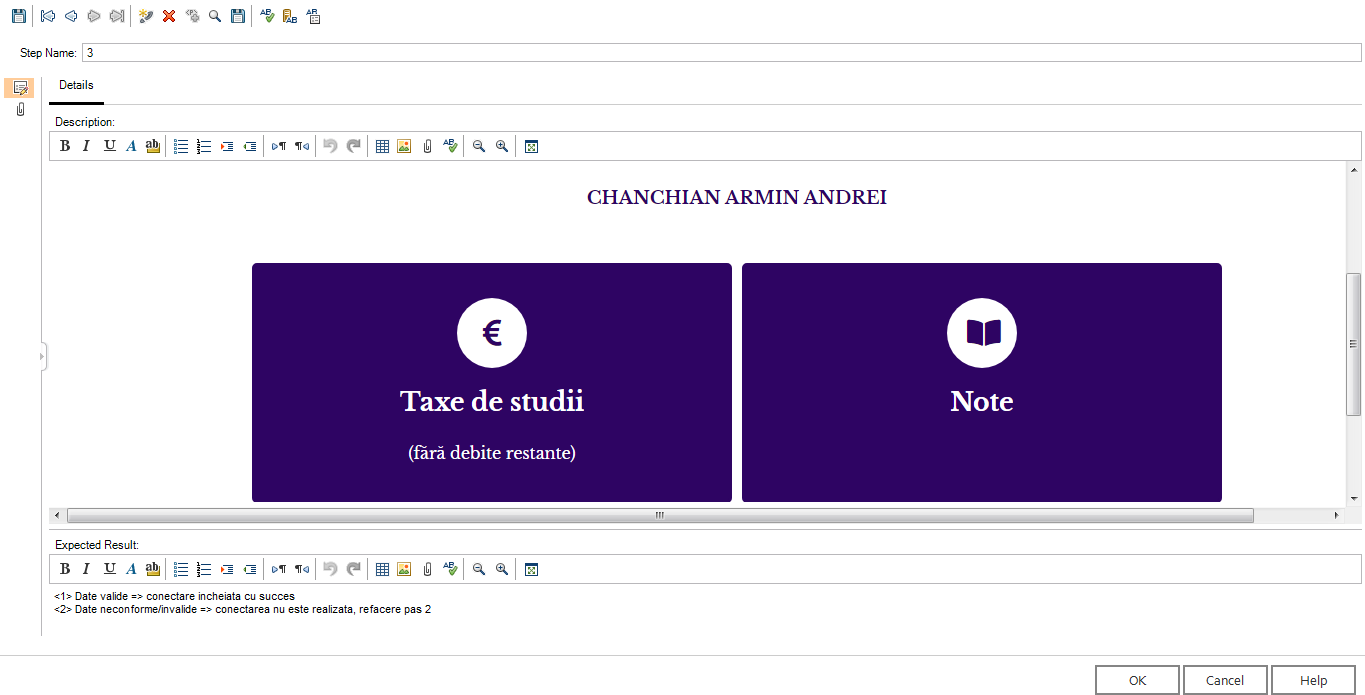
**Test Plan-ul** detaliaza planul de testare, inclusiv obiectivele, resursele necesare, metodele si tehnologiile de testare, planul de implementare si programul de testare. In aceasta sectiune sunt definiti pasii necesari pentru testarea fiecarui Requirement setat anterior.



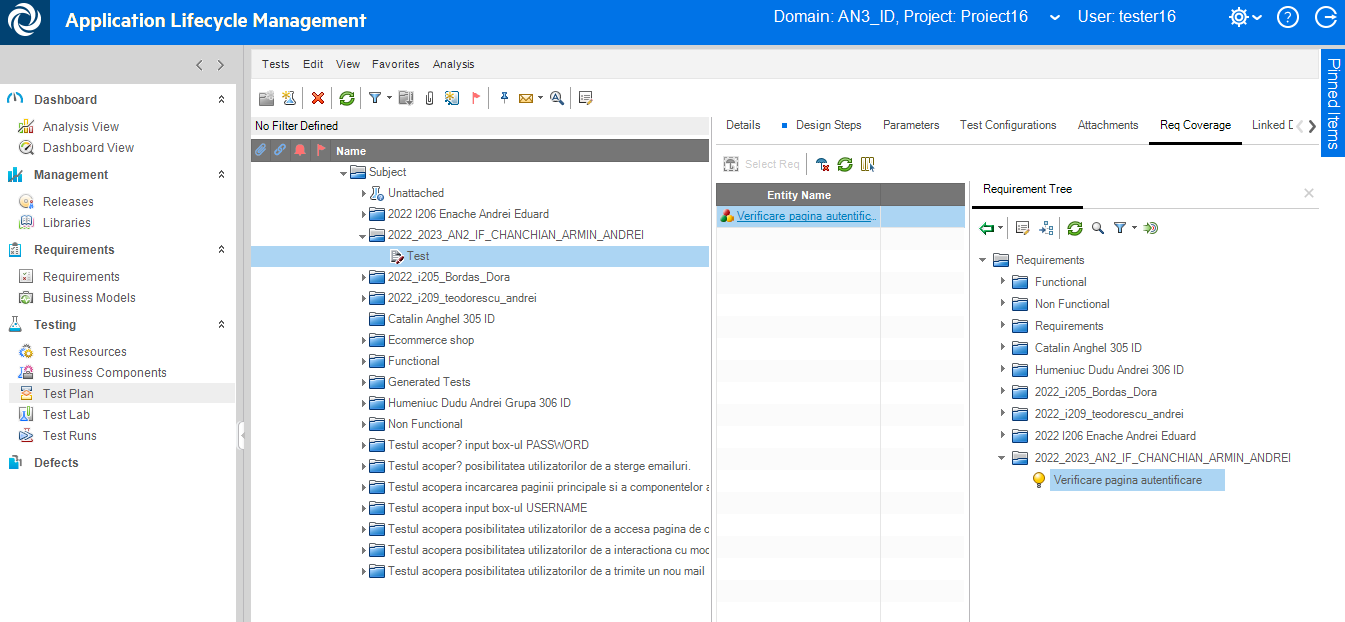
In **Design Steps** sunt definiti pasii:







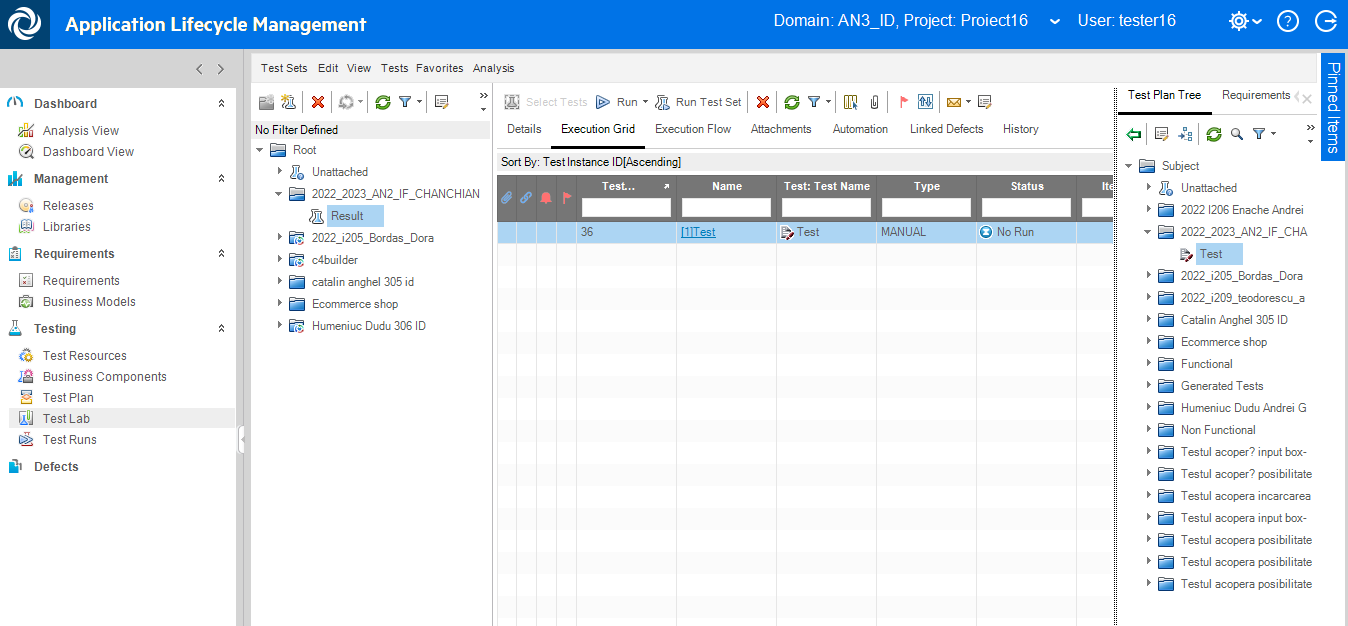
Pentru anexarea Design Steps cu Requirements, se acceseaza **Req Coverage** si se asociaza cerinta.



**7. Creare Test Lab**

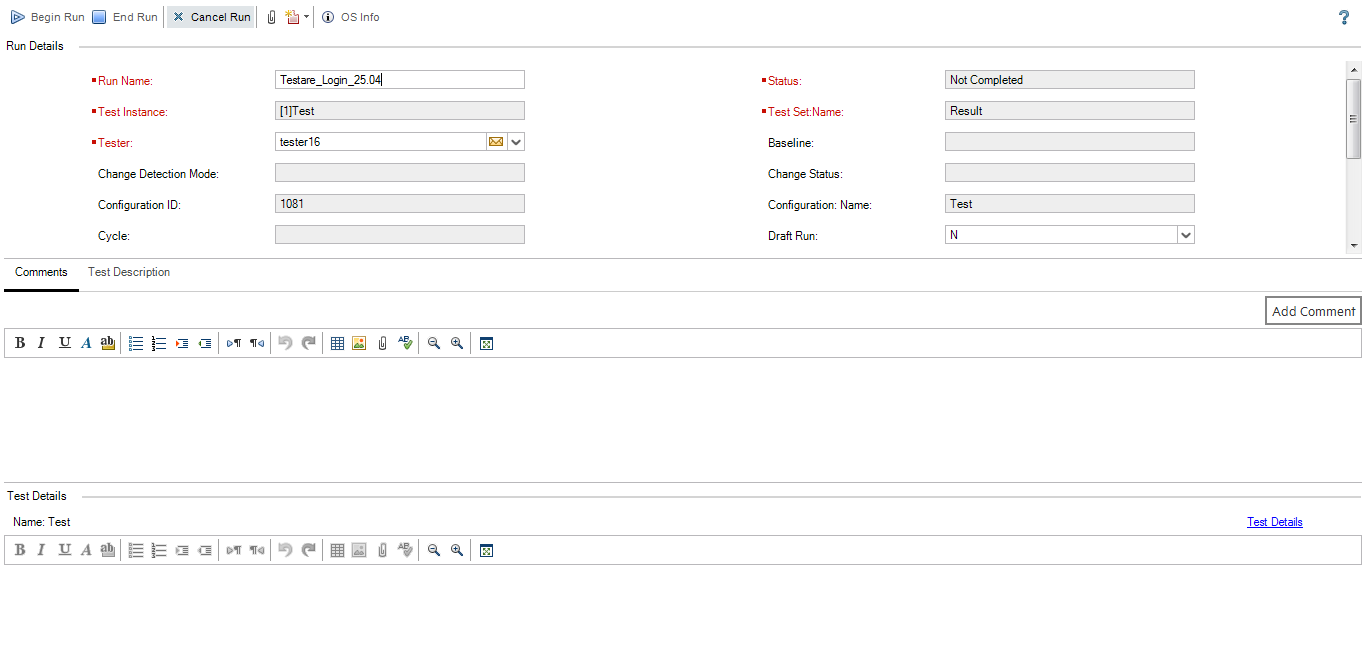
**Test Plan-ul** reprezinta un spatiu de lucru pentru echipa de testare, unde se pot organiza, executa si monitoriza activitatile de testare a produsului software. Test Lab-ul este compus din grupuri de teste (test sets), care contin un set de test cases si pot fi asociate cu diverse versiuni ale produsului software.

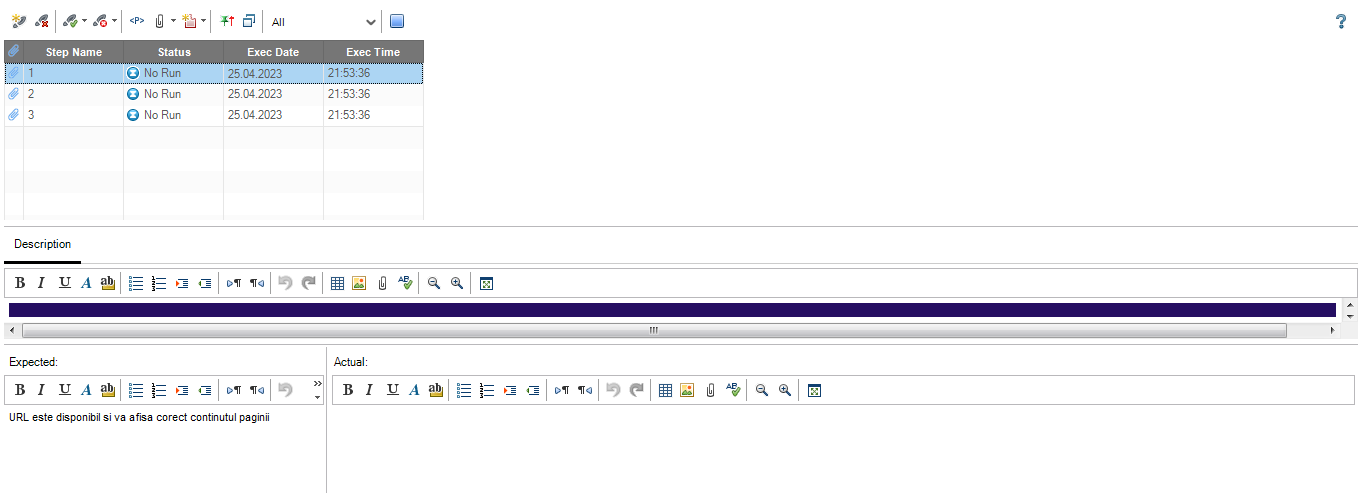
In urma crearii planului de teste, acestea pot fi conturate si marcate drept **PASSED** sau **FAILED**.



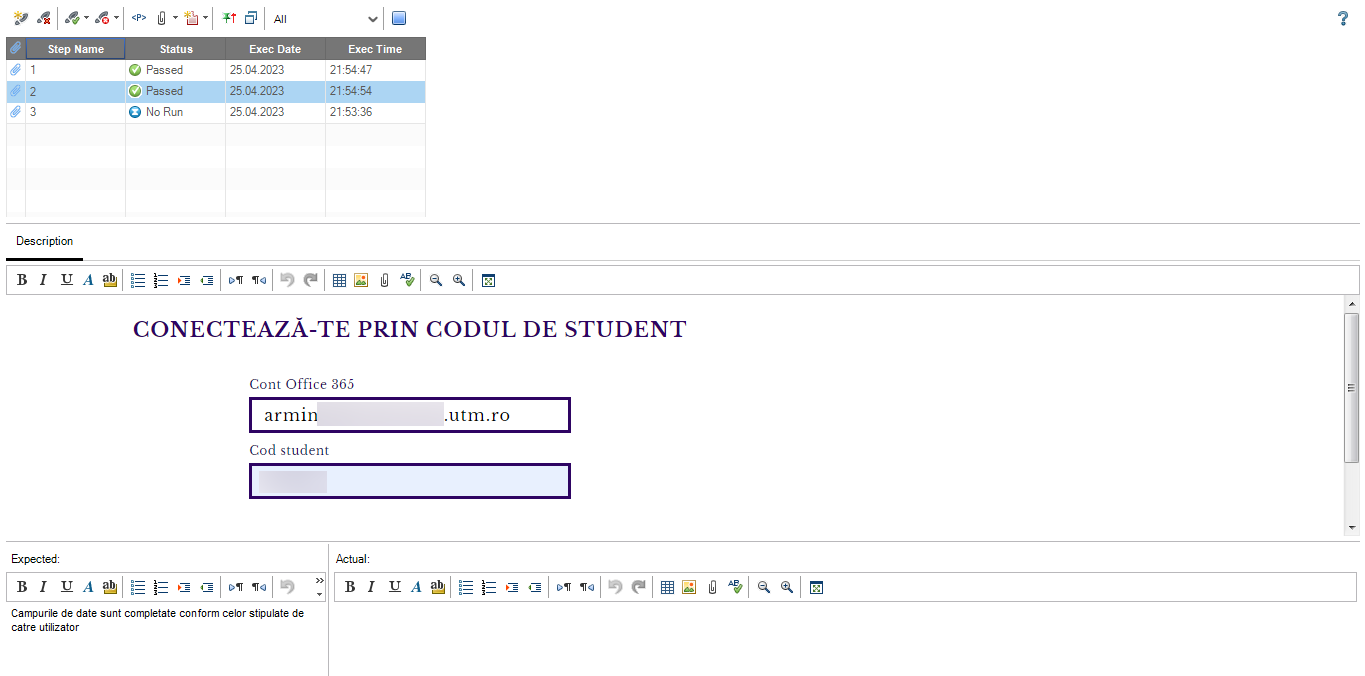
Apasarea functiei **Run**, cu planul selectat va duce la aparitia unei interfete care

concretizeaza detaliile respectivei incercari de testare, care va porni la apasarea **Begin Run**.



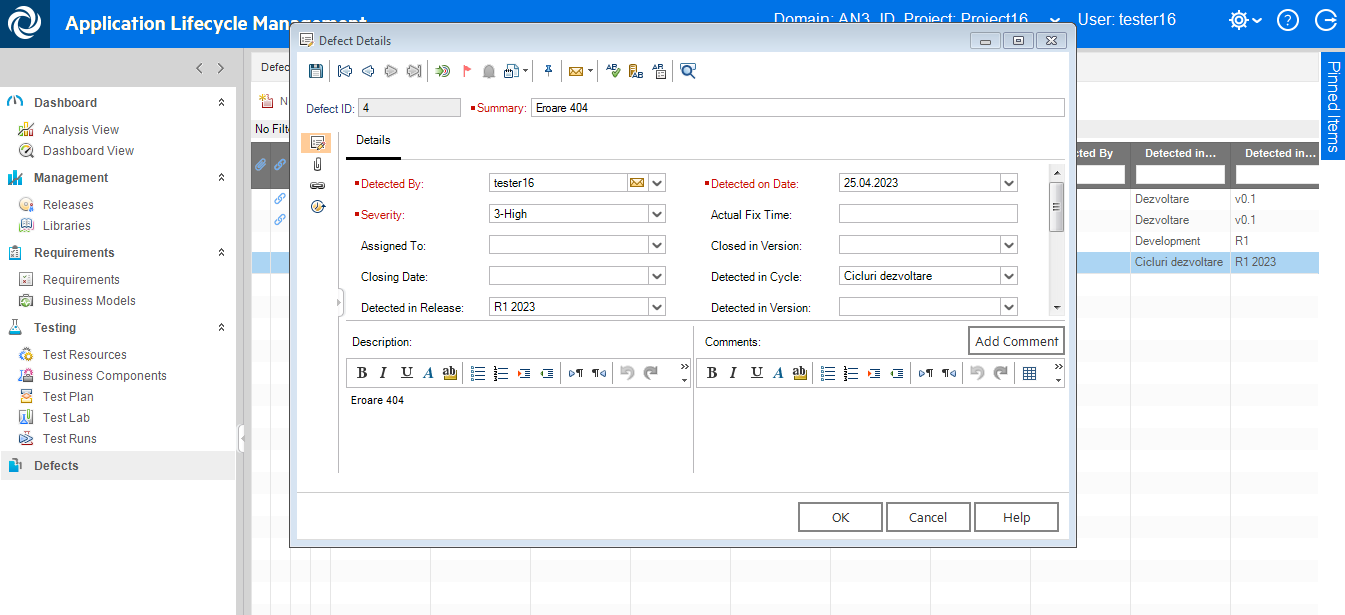


Pasii ce trebuie executati pot fi marcati drept **PASSED** ori dupa caz, **FAILED**. De asemenea, pot fi lasate explicatii in cazul in care testul a esuat.

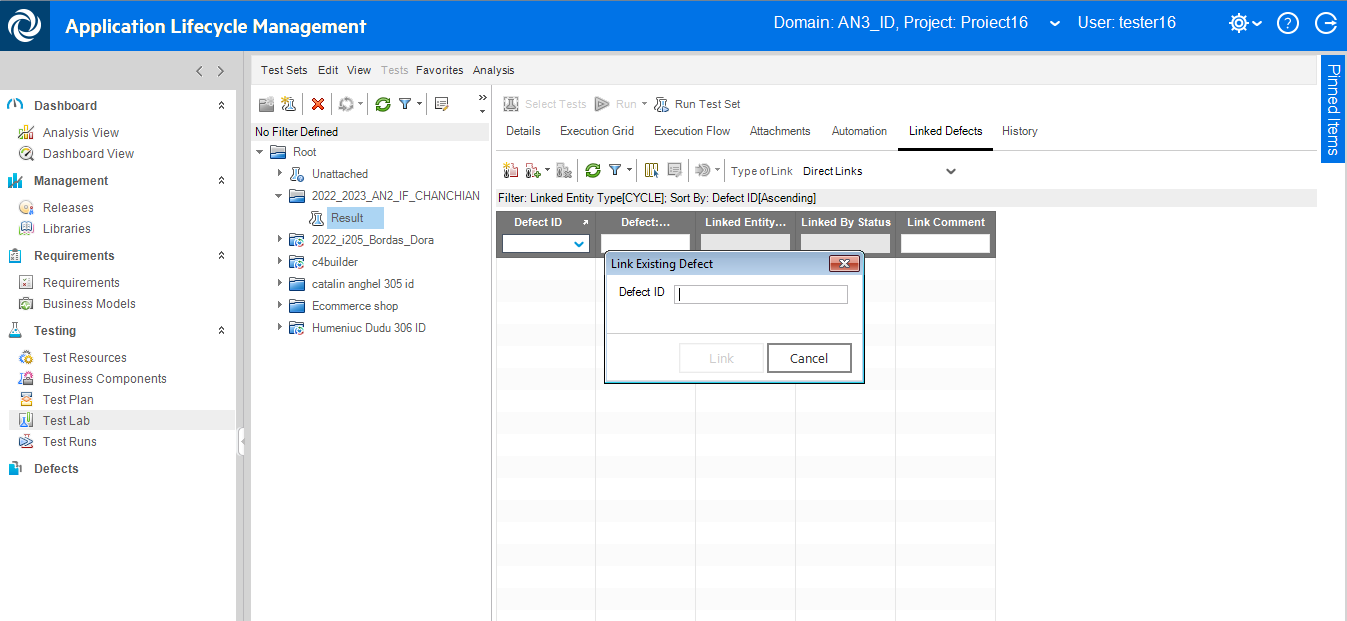


**8. Sesizarea Defects**

**Defects** reprezinta o sectiune a platformei dedicate gestionarii defectelor sau erorilor gasite in timpul procesului de testare a produsului software.

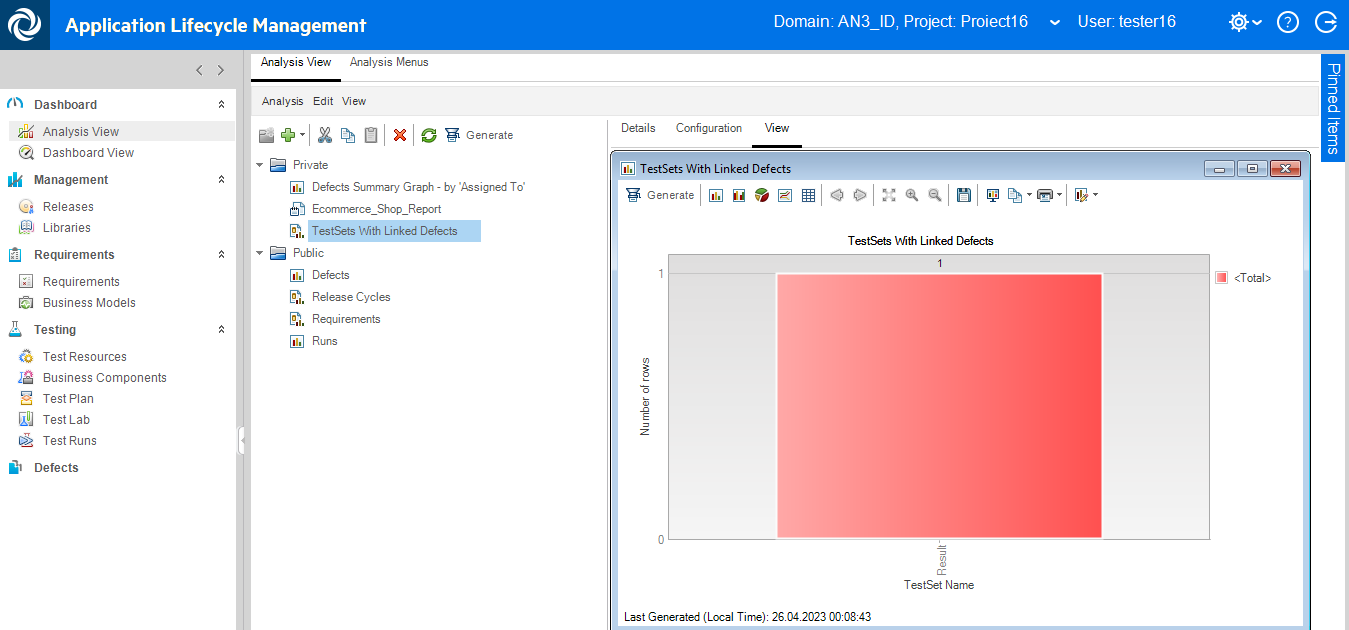


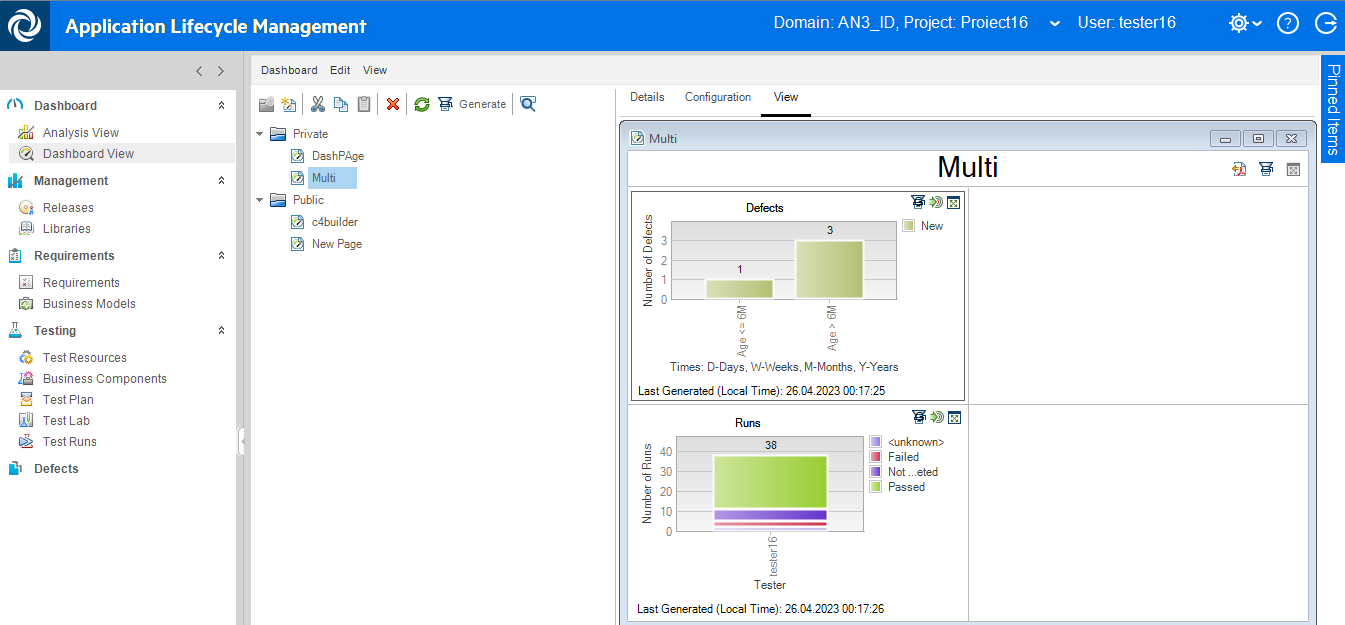
Defectele pot fi asociate in mod direct cu Test Set prin functia **Link Existing Defect**.



**9. Dashboard**

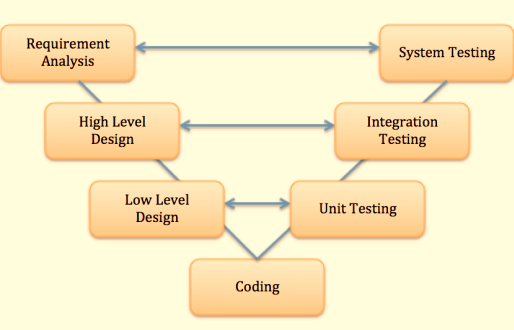
**Dashboard** este o functionalitate care permite utilizatorilor sa vizualizeze si sa gestioneze informatiile legate de proiect intr-un mod simplu si accesibil. Poate include diverse panouri si widget-uri care furnizeaza informatii despre diverse aspecte ale proiectului, cum ar fi starea testelor, starea defectelor, progresul proiectului, riscurile si problemele identificate.



Este constituit un raport in sectiunea Analysis View, prin folosirea functiei **New Item**.

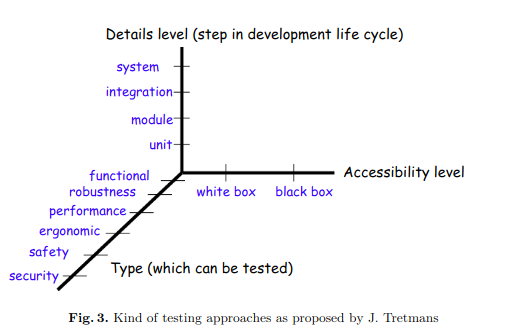
S-a initializat un bord cu multiple grafice in sectiunea **Dashboard View**, folosindu-se functia **New Page**.

## Principii generale



Software Devellopment Life Cycle (SDLC) vs Software Testing Life Cycle(STLC)

Clasificare teste:



## Intrebari

**1.** Ce este MicroFocus ALM si cum poate fi utilizat in procesul de dezvoltare de software?

Este o solutie software de management al ciclului de viata al aplicatiilor care ajuta la gestionarea procesului de dezvoltare si testare a aplicatiilor de la conceptie pana la livrare. Poate fi utilizat pentru planificarea si urmarirea testelor, gestionarea defectelor si raportarea progresului proiectului.

**2.** Cum poate ALM ajuta la imbunatatirea calitatii produselor software?

ALM-ul are o baza puternica, prin planificarea si urmarirea testelor, gestionarea defectelor si raportarea progresului. Aceasta permite echipei de dezvoltare sa identifice problemele mai devreme in ciclul de viata al aplicatiei, ceea ce poate duce la o reducere a costurilor si a timpului necesar pentru remedierea problemelor. De asemenea, rapoartele si analizele furnizate pot ajuta la identificarea tendintelor si la imbunatatirea proceselor de dezvoltare si testare pe termen lung.

**3.** Care sunt difrerentele intre testarea Black Box si White box?

Testarea tip black box este o metoda de testare a software-ului care se concentreaza pe functionalitatea si comportamentul acestuia, fara a lua in considerare codul sursa sau structura interna a programului. Cel ce testeaza nu are cunostinte despre cum este construit software-ul si se concentreaza doar pe intrari si iesiri, verificand daca functiile se comporta asa cum ar trebui sa se comporte. Testul black box este util pentru a verifica daca software-ul functioneaza corect din perspectiva utilizatorului.

Testarea tip white box este o metoda de testare a software-ului care se concentreaza pe structura interna a programului si pe logica codului sursa. Testerul are acces la codul sursa si cunostinte despre cum este construit software-ul, astfel incat poate efectua teste mai detaliate si poate identifica probleme in logica programului. Testul white box este util pentru a verifica acoperirea codului, performanta si optimizarea programului.