Software Testing of DSpace Digital Library Software

By

Mohammad Ehtesham

Guided by

Dr. A. R. D. Prasad

Associate Professor

Documentation Research & Training Centre

Project II

A guided research project submitted in partial fulfillment of the course leading to the Award of

"Associateship in Documentation and Information Science"

2005 -2007

Documentation Research and Training Centre

Indian Statistical Institute

Bangalore 560059

Software Testing of DSpace Digital Library Software

By

Mohammad Ehtesham

DECLARATION

I hereby declare that the project report entitled ***"Software Testing of DSpace Digital Library Software"*** which is being submitted in partial fulfillment of the course requirements leading to the award of *Associateship in Documentation and Information Science (ADIS)* in Documentation Research and Training Centre (DRTC), Indian Statistical Institute, Bangalore Center (ISIBC). This is the result of the work carried out by me under the guidance and supervision of ***Dr. A.R.D Prasad,*** DRTC.

I further declare that I or any other person has not previously submitted this project to any other institution/university for any other degree/diploma.

(Md Ehtesham)

It is certified that this project has been prepared under my guidance and supervision.

(Dr. A.R.D Prasad)

Place: Bangalore

Date: 30/08/07

ACKNOWLEDGEMENT

I take this opportunity to express my gratitude to my Mother and Father and to the almighty for giving me everything what I am now and for the ability to work.

I seize this opportunity to evince my earnest gratitude to my guide *Dr. A.R.D Prasad* for his complete guidance in every aspect and being such an ideal and an inspiring figure all along always propelling to work hard and to be contributive –I evince in him my reverence .

I thank *Prof. & Head Dr. I. K. Ravichandra Rao,* Documentation Research and Training Centre for his encouragement and guidance throughout the course. I pay my gratitude to *Prof. K.S. Raghavan* for his moral support and valuable suggestion. It is also an opportunity for me to pay my respect to *Dr. Devika P Madalli* for her support at every step in this course whenever I needed, and of course for this project her suggestions were invaluable. I thank her earnestly.

I am thankful to Ms. Sunita Barve, Library-In-Charge, National Centre for Radio Astrophysics for her invaluable suggestions and evaluation of this project, and the Research Scholars Ms. Nabonita Guha, Mr. Bishwanath Dutta and Mr. Anand Kumar Pandey for their suggestions and help in completion of this project.

At last, but by no means the least, I express my sincere thanks to all of my friends, colleagues and juniors who always have their best opinion for me, encouraging and helping out whenever it is needed. And, I can hardly forget their enthusiasm, appreciation, cooperation, assistance that has been a great encouragement to me.

Thank you all.

(Mohammad Ehtesham)

Contents

Chapter 1

Chapter 2

Chapter 3

Chapter 4

Chapter 5

Chapter 6

Chapter 1

1. Introduction

In today's ever-changing world of technology and information, a growing number of organizations and universities seek to store digital documents in an online, easily accessible manner. This is possible by establishing and maintaining a "Digital Library" in the institution. These "Digital Library" Repositories can be powerful systems that allow institutions to store and maintain their digital documents and allow interaction and collaboration among users in the organizations.

There are number of digital library software available in "Open Source" as well as in "Proprietary format". Before selecting any digital library software it is mandatory to plan and define specifications for our "Digital Library", and accordingly we can select software required to build up a Digital Repository, which meets our requirements both technically and functionally. In other words before implementing or using any software, its exhaustive testing is very much necessary.

In this project, testing of DSpace digital library software is undertaken. Before we see the detailed picture of technical and functional testing of DSpace, it is important to lay a common understanding or concept of Software Testing and DSpace overview.

1.1 Software Testing

Software Testing is the most important part of software development process. It is necessary to test software in order to develop and deliver a good, reliable and bug free software solution.

Software Testing can be defined as an “activity that helps in finding out bugs/defects/errors in a software system under development, in order to provide a bug free and reliable system/solution to the customer”.

In other words, we can consider an example as: suppose you are a good cook and are expecting some guests at dinner. You start making dinner; you make few very very delicious dishes (off-course, those which you already know how to make). And finally, when you are about to finish making the dishes, you ask someone (or you yourself) to check if everything is fine and there is no extra salt/chili/anything, which if is not in balance, can ruin your evening (This is what called 'TESTING').

This procedure you follow in order to make it sure that you do not serve your guests something that is not tasty! Otherwise your collar will go down and you will regret over your failure!

Software testing is both a discipline and a process. Though software testing is part of the software development process, it should not be considered part of software development. It is a separate discipline from software development. Software development is the process of coding functionality to meet defined end-user needs. Software testing is an iterative process of both validating functionality. The iterative process of software testing consists of:

- Designing tests
- Executing tests
- Identifying problems
- Getting problems fixed

The objective of software testing is to find problems and fix them to improve quality. Software testing typically represents 40% of a software development budget. Testing is the process of uncovering evidence of flaws and fixing these flaws in software systems. Flaws may result from various reasons such as mistakes, misunderstandings, and omissions occurring during any phase of

software development. Testing allows mitigating software risks. Testing is conducted according to a test plan, which is more effectively driven by the specific risks faced by the system. The risks determine what type of tests, needs to be conducted. The test plan determines how much testing is needed or is acceptable from a business perspective. Testing remains one of the most costly and challenging aspects of the software development process. From a business perspective, there is an optimum level of testing, which is acceptable. Beyond the optimum level, testing becomes less cost-effective, because the cost of testing simply exceeds the gains obtained from the defects detected.

1.2 DSpace Overview

DSpace (www.dspace.org) is a Digital Repository Software, created as a joint project between MIT Libraries and the Hewlett-Packard Company, and publicly released in November 2002 as Open-Source Software.

The DSpace Digital Repository software is freely available as open-source software from SourceForge (www.sourceforge.net/projects/dspace) under the terms of the BSD distribution license. Open-source software DSpace is available for anyone to download and run at any type of institution, organization, or company (or even just an individual). Users are also allowed to modify DSpace to meet an organization's specific needs. The specific terms of use are described in the BSD distribution license.

DSpace is one of the open source software platform to store, manage and distribute the collections in digital format. As much of the world's content is now being developed and disseminated in digital format, the DSpace software supports next-generation digital archiving that is more permanent and shareable than current analog archives. DSpace can support a wide variety of artifacts, including books, theses, and 3D digital scans of objects, photographs, film, video, research data sets and other forms of content.

DSpace was developed in response to expressed faculty needs for an easy-to-use, dependable service that could manage, host, preserve, and distribute faculty materials in digital formats. It offers faculty the advantages and convenience of web-based submission and dissemination. DSpace can accommodate a variety of genres like: (documents, datasets, and images) and formats like: (txt, pdf, doc, and jpeg).

It manages and distributes digital items, made up of digital files (or "bitstreams") and allows for the creation, indexing, and searching of associated metadata to locate and retrieve the items. It is also designed to support the long-term preservation of the digital material stored in the repository.

- For the user, it enables easy remote access and the ability to search and read DSpace items from one location: the World Wide Web.
- For the contributor, it offers the advantages of digital distribution and long-term preservation for a variety of types of documents including text, audio, video, images, datasets and more. Authors can store their digital works in collections that are maintained by DSpace "communities" within the parent institution, and these communities (such as university departments, laboratories, and research groups) can adapt the system to meet their individual needs and manage the submission process themselves.
- For the institution, it offers the opportunity to provide access to all the institution's research and teaching materials in digital format through one interface. The repository is organized to accommodate the varying policy and workflow issues inherent in a multi-disciplinary environment. Submission workflow and access policies can be customized to adhere closely to each community's needs.

DSpace differs from other digital library initiatives. It employs a distributed submission process and seeks to capture newly created digital research materials

in a broad range of formats. DSpace is also well suited to housing digitized historic collections to enhance the contextual reference for newly submitted works.

For the submission of research materials in DSpace, the self-defined, depositing Communities determines who may have access to archived works, with options ranging from a worldwide audience to a select few. There is no charge for submitting to or viewing digital material in DSpace.

DSpace provides a way to manage research materials and publications in a professionally maintained repository to give users greater visibility and accessibility over time.

DSpace enables organizations to:

- capture and describe digital material using a submission workflow module, or a variety of programmatic ingest options
- distribute an organization's digital assets over the web through search and retrieval system
- preserve digital assets over the long term

1.3 About this project

This project is undertaken as partial fulfillment of my two year ADIS course at DRTC, ISI. In this project I have studied, the concept of software testing, principles and different types of software testing such as: *Manual* and *Automated* testing, and done software testing for DSpace digital library software. I have studied the functionality of DSpace digital library Software and prepared test cases accordingly in order to test the functionality of Dspace which can be helpful in better understanding of this software as far its usage is concerned and find out the shortcomings and track the defects.

1.4 Objective of the Study

The objective of the present study is to evaluate and perform exhaustive software testing of Dspace digital library software in order, to layout better understanding of technical architecture and functionality of the software, which can be helpful in detailed specification and planning, to build up any digital library repository and ultimately accelerate the usage of the software.

1.5 Scope of the Project

Scope of this project is to study the functionality of DSpace digital library software, prepare the test cases based on functionality of the software, run the tests and tracks the defects if any.

1.6 Methodology

To perform software testing of DSpace digital library software various test cases have been prepared, run the tests and tracked the defects, based on functionality of the software using the automated software testing tool TestDirector of Mercury Interactive (Proprietary).

Mercury TestDirector allows deploying high-quality applications quickly and effectively by providing a consistent, repeatable process for requirements gathering, planning and scheduling tests, analyzing results, and managing defects and issues. TestDirector is a single, application for all essential aspects of test management — Requirements Management, Test Plan, Test Lab, and Defects Management.

1.7 Chapterization

Chapter 1

1 Introduction

1.1 Software Testing

1.2 Dspace overview

1.3 About this project

1.4 Objective of the Study

1.5 Scope of the Project

1.6 Methodology

1.7 Chapterization

Chapter 2

2 Software Testing Concepts and Need

2.1 Principle of Testing

2.2 Software Development Life Cycle

2.3 Software Development Life Cycle Model

Chapter 3

3 Software Testing Terms and Definitions

3.1 Verification and Validation

3.2 Project Management

3.3 Quality Management

3.4 Risk Management

3.5 Configuration Management

Chapter 4

Chapter 5

Chapter 6

Chapter 2

2. Software Testing Concepts and Need

Software testing is a critical element of software quality assurance and represents the ultimate process to ensure the correctness of the product. The quality product always enhances the customer confidence in using the product thereby increases the business economics. In other words, a good quality product means zero defects, which is derived from a better quality process in testing.

Software is an integrated set of Program codes, designed logically to implement a particular function or to automate a particular process. To develop a software product or project, user needs and constraints must be determined and explicitly stated. The development process is broadly classified into two.

1. Product development and
2. Project development

Product development is done assuming a wide range of customers and their needs. This type of development involves customers from all domains and collecting requirements from many different environments.

Project Development is done by focusing a particular customer's need, gathering data from his environment and bringing out a valid set of information that will help as a pillar to development process.

Testing is a necessary stage in the software life cycle: it gives the programmer and user some sense of correctness, though never "proof of correctness. With effective

testing techniques, software is more easily debugged, less likely to "break," more "correct", and, in summary, better.

Most development processes in the IT industry always seem to follow a tight schedule. Often, these schedules adversely affect the testing process, resulting in step motherly treatment meted out to the testing process. As a result, defects accumulate in the application and are overlooked so as to meet deadlines. The developers convince themselves that the overlooked errors can be rectified in subsequent releases.

Testing the product means adding value to it by raising the quality or reliability of the product. Raising the reliability of the product means finding and removing errors. Hence one should not test a product to show that it works; rather, one should start with the assumption that the program contains errors and then test the program to find as many of the errors as possible.

Definitions of Testing:

"Testing is the process of executing a program with the intent of finding errors"

Or

"Testing is the process of evaluating a system by manual or automatic means and verifies that it satisfies specified requirements"

Or

"... the process of exercising or evaluating a system or system component by manual or automated means to verify that it satisfies specified requirements or to identify differences / between expected and actual results..."

Why Software Testing?

Software testing helps to deliver quality software products that satisfy user's requirements, needs and expectations. If it is not done properly,

- Defects are found during operation,
- It results in high maintenance cost and user dissatisfaction

- It may cause mission failure
- Impact on operational performance and reliability

Software Bug: A Formal Definition

Calling any and all software problems bugs may sound simple enough, but doing so hasn't really addressed the issue. To keep from running in circular definitions, there needs to be a definitive description of what a bug is.

A software bug occurs when one or more of the following four rules are true:

1) The software doesn't do something that the product specification says it should do.
2) The software does something that the product specification says it shouldn't do.
3) The software does something that the product specification doesn't mention.
4) The software doesn't do something that the product specification doesn't mention but should.

2.1. Principles of Testing

The main objective of testing is to find defects in requirements, design, documentation, and code as early as possible. The test process should be such that the software product that will be delivered to the customer is defect less. All Tests should be traceable to customer requirements.

Test cases must be written for invalid and unexpected, as well as for valid and expected input conditions. A necessary part of a test case is a definition of the expected output or result. A good test case is one that has high probability of detecting an as-yet undiscovered error.

Eight Basic Principles of Testing:

- Define the expected output or result.

- Don't test your own programs.
- Inspect the results of each test completely.
- Include test cases for invalid or unexpected conditions.
- Test the program to see if it does what it is not supposed to do as well as what it is supposed to do.
- Avoid disposable test cases unless the program itself is disposable.
- Do not plan tests assuming that no errors will be found.

The probability of locating more errors in any one module is directly proportional to the number of errors already found in that module.

Best Testing Practices to be followed during testing:

- Testing and evaluation responsibility is given to every member, so as to generate team responsibility among all.
- Develop Master Test Plan so that resource and responsibilities are understood and assigned as early in the project as possible.
- Systematic evaluation and preliminary test design are established as a part of all system engineering and specification work.
- Testing is used to verify that all project deliverables and components are complete, and to demonstrate and track true project progress.
- A Risk prioritized list of test requirements and objectives (such as requirements-based, design-based, etc) are developed and maintained.
- Conduct Reviews as early and as often as possible to provide developer feedback and get problems found and fixed as they occur.

2.2. Software Development Life Cycle

Let us look at the Traditional Software Development life cycle *vs* presently or most commonly used life cycle.

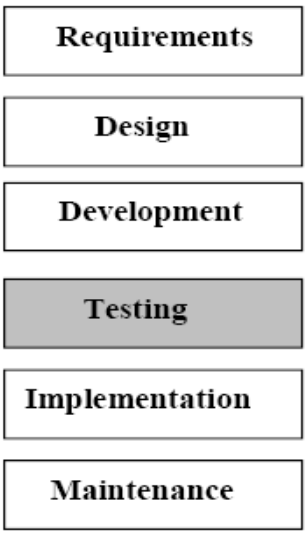

Fig A (Traditional)

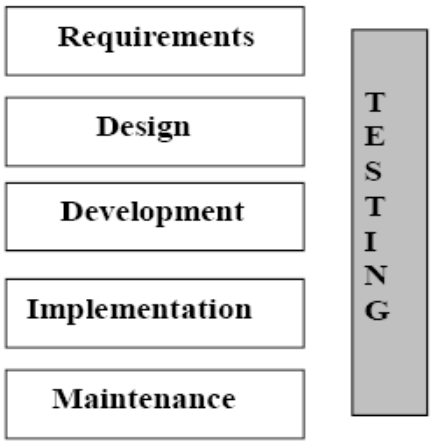

Fig B (Most commonly used)

In the above Fig A, the Testing Phase comes after the Development or coding is complete and before the product is launched and goes into Maintenance phase. We have some disadvantages using this model – Cost of fixing errors will be high because we are not able to find errors until coding is completed. If there is error at Requirements phase then all phases should be changed. So, total cost becomes very high.

The Fig B shows the recommended Test Process involves testing in every phase of the life cycle. During the Requirements phase, the emphasis is upon validation to determine that the defined requirements meet the needs of the organization. During Design and Development phases, the emphasis is on verification to ensure that the design and program accomplish the defined requirements. During the Test and Installation phases, the emphasis is on inspection to determine that the implemented system meets the system specification. During the maintenance phases, the system will be re-tested to determine that the changes work and that the unchanged portion continues to work.

Requirements and Analysis Specification

The main objective of the requirement analysis is to prepare a document, which includes all requirements of client. That is, the *Software Requirement Specification* (SRS) document is the primary output of this phase. Proper requirements and specifications are critical for having a successful project. Removing errors at this phase can reduce the cost as much as errors found in the Design phase. And also you should verify the following activities:

• Determine Verification Approach.
• Determine Adequacy of Requirements.
• Generate functional test data.
• Determine consistency of design with requirements.

Design phase

In this phase we are going to design entire project into two

• *High –Level Design or System Design.*
• *Low –Level Design or Detailed Design.*

High –Level Design or System Design (HLD)

High – level Design gives the overall System Design in terms of **Functional Architecture and Database design**.

This is very useful for the developers to understand the flow of the system. In this phase design team, review team (testers) and customers plays a major role. For this the entry criteria are the requirement document that is SRS. And the exit criteria will be HLD, projects standards, the functional design documents, and the database design document.

Low – Level Design (LLD)

During the detailed phase, the view of the application developed during the high level design is broken down into modules and programs. Logic design is done for

every program and then documented as **program specifications**. For every program, a **unit test** plan is created.

The entry criteria for this will be the HLD document. And the exit criteria will the program specification and unit test plan (LLD).

Development Phase

This is the phase where actually coding starts. After the preparation of HLD and LLD, the developers know what their role is and according to the specifications they develop the project. This stage produces the source code, executables, and database. The output of this phase is the subject to subsequent testing and validation. And we should also verify these activities:

• Determine adequacy of implementation.
• Generate structural and functional test data for programs.

The inputs for this phase are the physical database design document, project standards, program specification, unit test plan, program skeletons, and utilities tools. The output will be test data, source data, executables, and code reviews.

Testing phase

This phase is intended to find defects that can be exposed only by testing the entire system. Static Testing or Dynamic Testing can do this. Static testing means testing the product, which is not executing, we do it by examining and conducting the reviews. Dynamic testing is what you would normally think of testing. We test the executing part of the project. A series of different tests are done to verify that all system elements have been properly integrated and the system performs all its functions.

Implementation phase or the Acceptance phase

This phase includes two basic tasks:

• Getting the software accepted
• Installing the software at the customer site.

Acceptance consist of formal testing conducted by the customer according to the Acceptance test plan prepared earlier and analysis of the test results to determine whether the system satisfies its acceptance criteria. When the result of the analysis satisfies the acceptance criteria, the user accepts the software.

Maintenance phase

This phase is for all modifications, which is not meeting the customer requirements or any thing to append to the present system. All types of corrections for the project or product take place in this phase. The cost of risk will be very high in this phase. This is the last phase of software development life cycle. The input to this will be project to be corrected and the output will be modified version of the project.

2.3. Software Development Lifecycle Models

The process used to create a software product from its initial conception to its public release is known as the software development lifecycle model.

There are many different methods that can be used for developing software, and no model is necessarily the best for a particular project. There are four frequently used models:

- Big –Bang Model
- Waterfall Model
- Prototype Model
- Spiral Model

Big – Bang Model

The Big- Bang Model is the one in which we put huge amount of matter (people or money) is put together, a lot of energy is expended – often violently – and out comes the perfect software product or it doesn't.

The beauty of this model is that it's simple. There is little planning, scheduling, or Formal development process: All the effort is spent developing the software and writing the code. It's and ideal process if the product requirements aren't well understood and the final release date is flexible. It's also important to have flexible customers, too, because they won't know what they're getting until the very end.

Waterfall Model

A project using waterfall model moves down a series of steps starting from an initial idea to a final product. At the end of each step, the project team holds a review to determine if they're ready to move to the next step. If the project isn't ready to progress, it stays at that level until it's ready.

Each phase requires well-defined information, utilizes well-defined process, and results in welldefined outputs. Resources are required to complete the process in each phase and each phase is accomplished through the application of explicit methods, tools and techniques.

The Waterfall model is also called the Phased model because of the sequential move from one phase to another, the implication being that systems cascade from one level to the next in smooth progression. It has the following seven phases of development:

The figure: C represents the Waterfall Model.

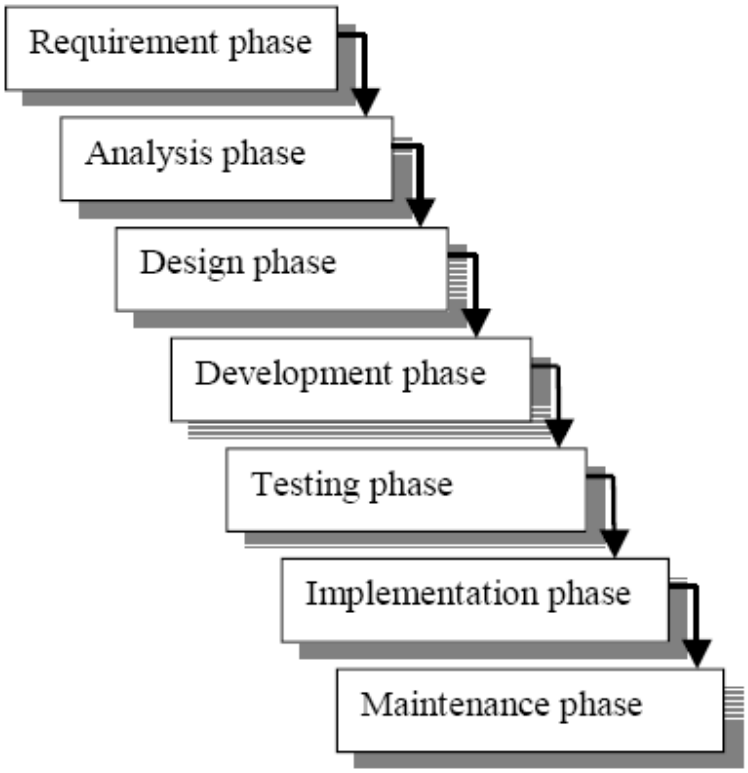

Figure: C Waterfall Model

Notice three important points about this model.

- There's a large emphasis on specifying what the product will be.
- The steps are discrete; there's no overlap.
- There's no way to back up. As soon as you're on a step, you need to complete the tasks for that step and then move on.

Prototype model

The Prototyping model, also known as the Evolutionary model, came into SDLC because of certain failures in the first version of application software. A failure in the first version of an application inevitably leads to need for redoing it. To avoid failure of SDLC, the concept of Prototyping is used.

The basic idea of Prototyping is that instead of fixing requirements before the design and coding can begin, a prototype is to understand the requirements. The

prototype is built using known requirements. By viewing or using the prototype, the user can actually feel how the system will work.

The prototyping model can been defined as:

"A model whose stages consist of expanding increments of an operational software with the direction of evolution being determined by operational experience."

Prototyping Process

The following activities are carried out in the prototyping process:

- The developer and die user work together to define the specifications of the critical parts of the system.
- The developer constructs a working model of the system.
- The resulting prototype is a partial representation of the system.
- The prototype is demonstrated to the user.
- The user identifies problems and redefines the requirements.
- The designer uses the validated requirements as a basis for designing the actual or production software

Prototyping is used in the following situations:

- When an earlier version of the system does not exist.
- When the user's needs are not clearly definable/identifiable.
- When the user is unable to state his/her requirements.
- When user interfaces are an important part of the system being developed.

Spiral model

The traditional software process models don't deal with the risks that may be faced during project development. One of the major causes of project failure in the past has been negligence of project risks. Due to this, nobody was prepared when something unforeseen happened. Barry Boehm recognized this and tried to incorporate the factor, project risk, into a life cycle model. The result is the Spiral

model, which was first presented in 1986. The new model aims at incorporating the strengths and avoiding the different of the other models by shifting the management emphasis to risk evaluation and resolution.

Each phase in the spiral model is split into four sectors of major activities.

These activities are as follows:

Objective setting:

This activity involves specifying the project and process objectives in terms of their functionality and performance.

Risk analysis:

It involves identifying and analyzing alternative solutions. It also involves identifying the risks that may be faced during project development.

Engineering:

This activity involves the actual construction of the system.

Customer evaluation:

During this phase, the customer evaluates the product for any errors and modifications.

Chapter 3

3 Software Testing Terms and Definitions

- Verification and validation
- Project Management
- Quality Management
- Risk Management
- Configuration Management
- Cost Management
- Compatibility Management

3.1 Verification and validation

Verification and validation are often used interchangeably but have different definitions. These differences are important in software testing.

Verification *is the process confirming that software meets its specifications.*

Validation *is the process confirming that it meets the user's requirements.*

Verification can be conducted through **Reviews**. Quality reviews provides visibility into the development process throughout the software development life cycle, and help teams determine whether to continue development activity at various checkpoints or milestones in the process. They are conducted to identify defects in a product early in the life cycle.

Types of Reviews

• **In-process Reviews:-**

They look at the product during a specific time period of life cycle, such as during the design activity. They are usually limited to a segment of a project, with the

goal of identifying defects as work progresses, rather than at the close of a phase or even later, when they are more costly to correct.

• **Decision-point or phase-end Reviews: -**

This type of review is helpful in determining whether to continue with planed activities or not. They are held at the end of each phase.

Post implementation Reviews: -

These reviews are held after implementation is complete to audit the process based on actual results. Post-implementation reviews are also know as "Postmortems", and are held to assess the success of the overall process after release and identify any opportunities for process improvements.

Classes of Reviews

• **Informal or Peer Review**: -

In this type of review generally a one-to one meeting between the author of a work product and a peer, initiated as a request for input regarding a particular artifact or problem. There is no agenda, and results are not formally reported.

These reviews occur as need-based through each phase of a project.

• **Semiformal or Walkthrough Review: -**

The author of the material being reviewed facilitates this. The participants are led through the material in one of the two formats: the presentation is made without interruptions and comments are made at the end, or comments are made throughout. Possible solutions for uncovered defects are not discussed during the review.

• **Formal or Inspection Review: -**

An inspection is more formalized than a 'walkthrough', typically with 3-8 people including a moderator, reader, and a recorder to take notes. The subject of the

inspection is typically a document such as a requirements spec or a test plan, and the purpose is to find problems and see what's missing, not to fix anything.

Attendees should prepare for this type of meeting by reading thru the document; most problems will be found during this preparation. The result of the inspection meeting should be a written report. Thorough preparation for inspections is difficult, painstaking work, but is one of the most cost effective methods of ensuring quality.

Three rules should be followed for all reviews:

1. The product is reviewed, not the producer.
2. Defects and issues are identified, not corrected.
3. All members of the reviewing team are responsible for the results of the review.

3.2 Project Management

Project management is Organizing, Planning and Scheduling software projects. It is concerned with activities involved in ensuring that software is delivered on schedule and in accordance with the requirements of the organization developing and procuring the software. Project management is needed because software development is always subject to budget and schedule constraints that are set by the organization developing the software.

Project management activities include:
• Project planning.
• Project scheduling.
• Iterative Code/Test/Release Phases
• Production Phase
• Post Mortem

Project planning

This is the most time-consuming project management activity. It is a continuous activity from initial concept through to system delivery. Project Plan must be regularly updated as new information becomes available. With out proper plan, the development of the project will cause errors or it may lead to increase the cost, which is higher than the schedule cost review.

Project scheduling

This activity involves splitting project into tasks and estimate time and resources required to complete each task. Organize tasks concurrently to make optional use of workforce. Minimize task dependencies to avoid delays caused by one task waiting for another to complete. Project Manager has to take into consideration various aspects like scheduling, estimating manpower resources, so that the cost of developing a solution is within the limits. Project Manager also has to allow for contingency in planning.

Iterative Code/Test/Release Phases

After the planning and design phases, the client and development team has to agree on the feature set and the timeframe in which the product will be delivered. This includes iterative releases of the product as to let the client see fully implemented functionality early and to allow the developers to discover performance and architectural issues early in the development. Each iterative release is treated as if the product were going to production. Full testing and user acceptance is performed for each iterative release. Experience shows that one should space iterations at least 2 – 3 months a part. If iterations are closer than that, more time will be spent on convergence and the project timeframe expands. During this phase, code reviews must be done weekly to ensure that the developers are delivering to specification and all source code is put under source control. Also, full installation routines are to be used for each iterative release, as it would be done in production.

Deliverables

• Triage

• Weekly Status with Project Plan and Budget Analysis

• Risk Assessment

• System Documentation

• User Documentation (if needed)

• Test Signoff for each iteration

• Customer Signoff for each iteration.

Production Phase

Once all iterations are complete, the final product is presented to the client for a final signoff. Since the client has been involved in all iterations, this phase should go very smoothly.

Deliverables

• Final Test Signoff

• Final Customer Signoff

Post Mortem Phase

The post mortem phase allows stepping back and reviewing the things that went well and the things that need improvement. Post mortem reviews cover processes that need adjustment, highlight the most effective processes and provide action items that will improve future projects.

To conduct a post mortem review, announce the meeting at least a week in advance so that everyone has time to reflect on the project issues they faced. Everyone has to be asked to come to the meeting with the following:

1. Items that were done well during the project
2. Items that were done poorly during the project

3. Suggestions for future improvements

During the meeting, collection of the information listed above is required. As each person offers their input, categorize the input so that all comments are collected. This will allow one to see how many people had the same observations during the project. At the end of observation review, a list of the items will be available that were mentioned most often. The list of items allowing the team to prioritize the importance of each item has to be perused. This will allow drawing a distinction of the most important items. Finally, a list of action items has to be made that will be used to improve the process and publish the results. When the next project begins, everyone on the team should review the Post Mortem Report from the prior release as to improve the next release.

3.3 Quality Management

The project quality management knowledge area is comprised of the set of processes that ensure the result of a project meets the needs for which the project was executed. Processes such as quality planning, assurance, and control are included in this area. Each process has a set of input and a set of output. Each process also has a set of tools and techniques that are used to turn input into output.

Definition of Quality:

• ***Quality*** *is the totality of features and characteristics of a product or service that bare on its ability to satisfy stated or implied needs.*

Or

• ***Quality*** *is defined as meeting the customer's requirement for the first time and for every time. This is much more that absence of defects which allows us to meet the requirements.*

Some goals of quality programs include:

• Fitness for use. *(Is the product or service capable of being used?)*

• Fitness for purpose. *(Does the product or service meet its intended purpose?)*

• Customer satisfaction. *(Does the product or service meet the customer's expectations?)*

Quality Management Processes

Quality Planning:

The Quality Planning is the process of identifying which quality standards is relevant to the project and determining how to satisfy them.

• ***Input includes****: Quality policy, scope statement, product description, standards and regulations, and other process Output.*

• ***Methods used****: benefit / cost analysis, benchmarking, flowcharting, and design of experiments.*

• ***Output includes****: Quality Management Plan, operational definitions, checklists, and Input to other processes.*

Quality Assurance

The Quality Assurance is the process of evaluating overall projects performance on a regular basis to provide confidence that the project will satisfy the relevant quality standards.

• ***Input includes****: Quality Management Plan, results of quality control measurements, and operational definitions.*

• ***Methods used****: quality planning tools and techniques and quality audits.*

• ***Output includes****: quality improvement.*

Quality Control

The Quality Control is the process of monitoring specific project results to determine if they comply with relevant quality standards and identifying ways to eliminate causes of unsatisfactory performance.

• ***Input includes****: work results, Quality Management Plan, operational definitions, and checklists.*

• ***Methods used include****: inspection, control charts, Pareto charts, statistical sampling, flowcharting, and trend analysis.*

• ***Output includes****: quality improvements, acceptance decisions, rework, completed checklists, and process adjustments.*

Quality Policy

The overall quality intentions and direction of an organization as regards quality, as formally expressed by top management

Total Quality Management (TQM)

A common approach to implementing a quality improvement program within an organization

Quality Concepts

• Zero Defects

• The Customer is the Next Person in the Process

• Do the Right Thing Right the First Time *(DTRTRTFT)*

• Continuous Improvement Process *(CIP) (From Japanese word, Kaizen)*

Tools of Quality Management

Problem Identification Tools:

• **Pareto Chart**

1. Ranks defects in order of frequency of occurrence to depict 100% of the defects. (Displayed as a histogram)

2. Defects with most frequent occurrence should be targeted for corrective action.

3. 80-20 rule: 80% of problems are found in 20% of the work.

4. Does not account for severity of the defects

• **Cause and Effect Diagrams (fishbone diagrams or Ishikawa diagrams)**

1. Analyzes the Input to a process to identify the causes of errors.

2. Generally consists of 8 major Input to a quality process to permit the characterization of each input.

• **Histograms**

1. Shows frequency of occurrence of items within a range of activity.

2. Can be used to organize data collected for measurements done on a product or process.

• **Scatter diagrams**

1. Used to determine the relationship between two or more pieces of corresponding data.

2. The data are plotted on an "X-Y" chart to determine correlation (highly positive, positive, no correlation, negative, and highly negative)

Problem Analysis Tools

1. Graphs
2. Check sheets (tic sheets) and check lists
3. Flowcharts

3.4 Risk Management

Risk management must be an integral part of any project. Everything does not always happen as planned. Project risk management contains the processes for identifying, analyzing, and responding to project risk. Each process has a set of input and a set of output. Each process also has a set of tools and techniques that are used to turn the input into output

Risk Management Processes

Risk Management Planning

Used to decide how to approach and plan the risk management activities for a project.

• ***Input includes***: The project charter, risk management policies, and ***WBS*** all serve as input to this process

• ***Methods used***: Many planning meeting will be held in order to generate the Risk Management plan

• ***Output includes***: The major output is the risk management plan, which does not include the response to specific risks. However, it does include methodology to be used, budgeting, timing, and other information

Risk Identification

Determining which risks might affect the project and documenting their characteristics

• ***Input includes***: The risk management plan is used as input to this process

• ***Methods used***: Documentation reviews should be performed in this process. Diagramming techniques can also be used

Output includes: Risk and risk symptoms are identified as part of this process. There are generally two types of risks. They are business risks that are risks of gain or loss. Then there are pure risks that represent only a risk of loss. Pure risks are also known as insurable risks

Risk Analysis

A qualitative analysis of risks and conditions is done to prioritize their affects on project objectives.

• ***Input includes***: There are many items used as input into this process. They include things such as the risk management plan. The risks should already be identified as well. Use of low precision data may lead to an analysis that is not useable. Risks are rated against how they impact the projects objectives for cost, schedule, scope, and quality

• ***Methods used***: Several tools and techniques can be used for this process. Probability and Impact will have to be evaluated

• ***Output includes***: An overall project risk ranking is produced as a result of this process. The risks are also prioritized. Trends should be observed. Risks calculated as high or moderate are prime candidates for further analysis

Risk Monitoring and Control

Used to monitor risks, identify new risks, execute risk reduction plans, and evaluate their effectiveness throughout the project life cycle.

• ***Input includes***: Input to this process includes the risk management plan, risk identification and analysis, and scope changes

• ***Methods used***: Audits should be used in this process to ensure that risks are still risks as well as discover other conditions that may arise.

• ***Output includes***: Output includes work-around plans, corrective action, project change requests, as well as other items

Risk Management Concepts

Expected Monetary Value (EMV)

• A Risk Quantification Tool

• EMV is the product of the risk event probability and the risk event value

• Risk Event Probability: An estimate of the probability that a given risk event will occur

Decision Trees

A diagram that depicts key interactions among decisions and associated chance events as understood by the decision maker. Can be used in conjunction with EMV since risk events can occur individually or in groups and in parallel or in sequence.

3.5 Configuration Management

Configuration management (CM) is the processes of controlling, coordinate, and tracking the Standards and procedures for managing changes in an evolving software product. Configuration Testing is the process of checking the operation of the software being tested on various types of hardware. Configuration management involves the development and application of procedures and standards to manage an evolving software product. This can be seen as part of a more general quality management process. When released to CM, software systems are sometimes called *baselines*, as they are a starting point for further development. The best bet in this situation is for the testers to go through the process of reporting whatever bugs or blocking-type problems initially show up, with the focus being on critical bugs. Since this type of problem can severely affect schedules, and indicates deeper problems in the software development process (such as insufficient unit testing or insufficient integration testing, poor design, improper build or release procedures, etc.) managers should be notified, and provided with some documentation as evidence of the problem.

Configuration management can be managed through

• Version control.
• Changes made in the project.

Version Control and Release management

Version is an instance of system, which is functionally distinct in some way from other system instances. It is nothing but the updated or added features of the previous versions of software. It has to be planned as to when the new system version is to be produced and it has to be ensured that version management procedures and tools are properly applied.

Release is the means of distributing the software outside the development team. Releases must incorporate changes forced on the system by errors discovered by

users and by hardware changes. They must also incorporate new system functionality.

Changes made in the project

This is one of most useful way of configuring the system. All changes will have to be maintained that were made to the previous versions of the software. This is more important when the system fails or not meeting the requirements. By making note of it one can get the original functionality. This can include documents, data, or simulation.

Configuration Management Planning

This starts at the early phases of the project and must define the documents or document classes, which are to be managed. Documents, which might be required for future system maintenance, should be identified and included as managed documents. It defines

- The types of documents to be managed
- Document-naming scheme
- Who takes responsibility for the CM procedures and creation of baselines
- Polices for change control and version management.

This contains three important documents they are

• Change management items.

• Change request documents.

• Change control board. (CCB)

Change management

Software systems are subject to continual change requests from users, from developers, from market forces. Change management is concerned with keeping, managing of changes and ensuring that they are implemented in the most cost-effective way.

Change request form

Definition of change request form is part of CM planning process. It records changes required, reason "why change -was suggested and urgency of change (from requestor of the change). It also records change evaluation, impact analysis, change cost and recommendations (System maintenance staff). A major problem in change management is tracking change status. Change tracking tools keep track the status of each change request and automatically ensure that change requests are sent to the right people at the right time, integrated with Email systems, allowing electronic change request distribution system.

Change control board

A group, who decide, whether or not they are cost-effective from a strategic, organizational and technical viewpoint, should review the changes. This group is sometimes called a change control board and includes members from project team.

Chapter 4

4 Types of Software Testing

Software testing consists of several subcategories, each of which is done for different purposes, and often using different techniques. Software testing categories include:

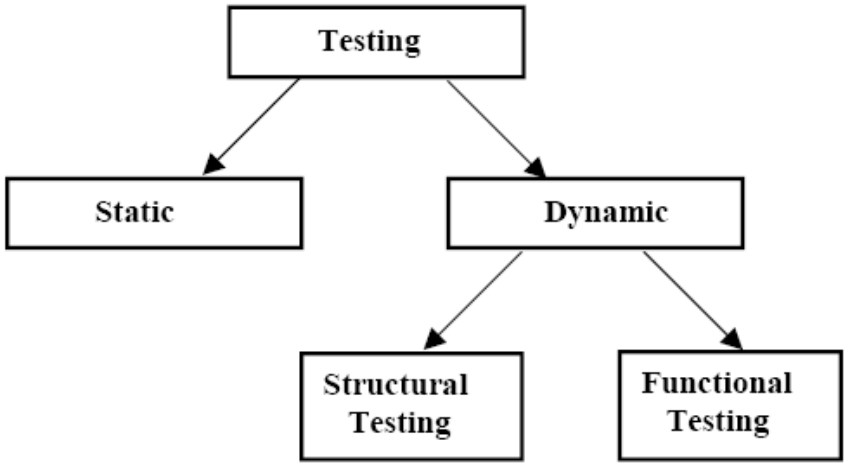

Figure: D Types of software testing

4.1 Static Testing

Static testing refers to testing something that's not running. It is examining and reviewing it. For e.g.: The specification of any software is a document and not an executing program, so it is considered as static. It is also something that was created using written or graphical documents or a combination of both.

In Static testing specification of the software under development is reviewed and tested in order to meet or check all the functions and feature specified in the specification document of the respective software.

High-level Reviews of specification

- Pretend to be the customer – Reviews is done to meet all the intended customers' needs and specifications.
- Research existing Standards and Guidelines – To check the development process is at par with or following the existing standards and guidelines or not like: CMM, CMMI, Six Sigma etc.
- Review and Test similar software – "Survival is the fittest" To exist and incorporate the features of other similar software review and check list is done through comparative review and testing of other similar software.

Low-level Reviews of specification

- Specification Attributes checklist – All listed out features and functionality of the software is checked.
- Specification terminology checklist – Various technical aspects of the software under development is tested and checked.

4.2 Dynamic Testing

Dynamic testing has two parts:

- Structural (usually called "white box") testing.
- Functional ("black box") testing.

Structural testing or White box testing

Structural tests verify the structure of the software itself and require complete access to the source code. This is also known as 'white box' testing because it allows test the internal workings of the code.

White-box tests make sure that the software structure itself contributes to proper and efficient program execution. Complicated loop structures, common data areas, Well-designed control structures, sub-routines and reusable modular programs are good or not.

White-box testing strength is also its weakness because some times it becomes too complicated and time taking, it is not very much cost effective. The source-code of the software needs to be examined by highly skilled technicians. That means that tools and skills required are highly specialized to the particular language and environment. Also, large or distributed system execution goes beyond one program, so a correct procedure might call another program that provides bad data. In large systems, it is the execution path as defined by the program calls, their input and output and the structure of common files that is important. This gets into a hybrid kind of testing that is often employed in intermediate or integration stages of testing.

Functional or Black Box Testing

Functional tests examine the behavior of software as evidenced by its output without reference to internal functions. Hence it is called ‘black box’ testing. If the program consistently provides the desired features with acceptable performance, then specific source code features are irrelevant (not taken into consideration). It's a pragmatic and down-to-earth assessment of software.

Functional or Black box tests address in a better way the modern programming paradigm. As object oriented programming, automatic code generation and code re-use becomes more prevalent, analysis of source code itself becomes less important and functional tests are becoming more important.

Black box tests also better attack the quality target. Since only the people paying for an application can determine if it meets their needs or not, it is an advantage to create the quality criteria from this point of view from the beginning of the testing exercise.

Black box tests have a basis in the scientific method. Like the process of science, Black box tests must have a hypothesis (specifications), a defined method or procedure (test plan), reproducible components (test data), and a standard notation

to record the results. One can re-run black box tests after a change to make sure the change only produced intended results with no inadvertent effects.

4.3 Testing levels

There are several types of testing in a comprehensive software test process, many of which occur simultaneously.

- Unit Testing
- Integration Testing
- System Testing
- Performance / Stress Testing
- Regression Testing
- Quality Assurance Testing
- User Acceptance Testing and Installation Testing

Unit Testing

Testing each module individually is called Unit Testing. This follows a White-Box testing concept. In some organizations, a peer review panel performs the design and/or code inspections. Unit or component tests usually involve some combination of structural and functional tests by programmers in their own systems. Unit tests often require building some kind of supporting framework (language or system environment) that allows units or components to execute.

Integration testing

The individual components are combined with other components to make sure that necessary communications, links and data sharing occur properly. It is not truly system testing because the components are not implemented in the operating environment. The integration phase requires more planning and some reasonable sub-set of production-type data. Larger systems often require several integration steps.

There are three basic integration test methods:

• All-at-once

• Bottom-up

• Top-down

The **all-at-once** method provides a useful solution for simple integration problems, involving a small program possibly using a few previously tested modules.

Bottom-up testing involves individual testing of each module using a driver routine that calls the module and provides it with needed resources. Bottom-up testing often works well in less structured manner, because there is less dependency on availability of other resources to accomplish the test. It is a more intuitive approach to testing that also usually finds errors in critical routines earlier than the top-down method. However, in a new system many modules must be integrated to produce system-level behavior, thus interface errors surface late in the process.

Top-down testing fits a prototyping environment that establishes an initial skeleton that fills individual modules that is completed. The method lends itself to more structured organizations that plan out the entire test process. Although interface errors are found earlier, errors in critical low-level modules can be found later than you would like.

System Testing

The system test phase begins once modules are integrated enough to perform tests in a whole system environment. System testing can occur in parallel with integration test, especially with the top-down method.

Performance / Stress Testing

An important phase of the system testing, often-called load or volume or performance/stress tests tries to determine the failure point of a system under extreme pressure. Stress tests are most useful when systems are being scaled up to

larger environments or being implemented for the first time. Web sites, or like any other large-scale multi-user system that requires multiple accesses and processing, contain vulnerable nodes that should be tested before deployment. Unfortunately, most stress testing can only simulate loads on various points of the system and cannot truly stress the entire network, as the users would experience it. Fortunately, once stress and load factors have been successfully overcome, it is only necessary to stress test again if major changes take place. A drawback of performance testing is it confirms the system can handle heavy loads, but cannot so easily determine if the system is producing the correct result or not.

Regression Testing

Regression tests confirm that implementation of changes have not adversely affected other functions. Regression testing is a type of test which is not supposed to be phase in a testing process. Regression tests apply at all phases whenever a change is made.

Quality Assurance Testing

Some organizations maintain a Quality Group that provides a different point of view, uses a different set of tests, and applies the tests in a different, more complete test environment. The group might look to see that organization standards have been followed in the specification, coding and documentation of the software. They might check to see that the original requirement is documented, verify that the software properly implements the required functions, and see that everything is ready for the users to take hands on experience at it.

User Acceptance Test and Installation Testing

Traditionally, this is where the users ‘get their first crack’ at the software. Unfortunately, by this time, it's usually too late. If the users have not seen prototypes, been involved with the design, and understood the evolution of the system, they are inevitably going to be unhappy with the result. If one can

perform every test as user acceptance tests, there is much better chance of a successful project.

4.4 Types of Testing Techniques

White Box Testing Technique

White box testing examines the basic program structure and it derives the test data from the program logic, ensuring that all statements and conditions have been executed at least once. White box tests verify that the software design is valid and also whether it was built according to the specified design.

Different methods used are:
Statement coverage – executes all statements at least once. (each and every line)
Decision coverage – executes each decision direction at least once.
Condition coverage – executes each and every condition in the program with all possible outcomes at least once.

Black Box Testing Technique

Black-box test technique treats the system as a "black-box", so it doesn't explicitly use knowledge of the internal structure. Black-box test design is usually described as focusing on testing functional requirements. Synonyms for black box include: Behavioral, Functional, Opaque-box, and Closed-box.

Black box testing is conducted on integrated, functional components whose design integrity has been verified through completion of traceable white box tests. Black box testing traces the requirements focusing on system externals. It validates that the software meets the requirements irrespective of the paths of execution taken to meet each requirements.

Three successful techniques for managing the amount of input data required includes:

- *Equivalence Partitioning*
- *Boundary Analysis*
- *Error Guessing*

Equivalence Partitioning:

Equivalence partitioning is the process of methodically reducing the huge (infinite) set of possible test cases into a much smaller, but still equally effective set. An Equivalence class is a subset of data that is representative of a larger class. Equivalence partitioning is a technique for testing equivalence classes rather than undertaking exhaustive testing of each value of the larger class, when looking for equivalence partitions, thinks about ways to group similar inputs, similar outputs, and similar operations of the software. These groups are the equivalence partitions.

Boundary value analysis:

If one can safely and confidently walk along the edge of a cliff without falling off, he can almost certainly walk in the middle of a field. If software can operate on the edge of its capabilities, it will almost certainly operate well under normal conditions. This technique consists of developing test cases and data that focus on the input and output boundaries of a given function.

Error Guessing

This is based on the theory that test cases can be developed based upon the intuition and experience of the Test-Engineer.

Incremental testing

Incremental testing is a disciplined method of testing the interfaces between unit-tested programs as well as between system components. It involves adding unit-tested programs to a given module or component one by one, and testing each result and combination.

There are two types of incremental testing:

Top-down: - This begins testing from top of the module hierarchy and work down to the bottom using interim stubs to simulate lower interfacing modules or programs. Modules are added in descending hierarchical order.

Bottom-up: - This begins testing from the bottom of the hierarchy and works up to the top. Modules are added in ascending hierarchical order. Bottom-up testing requires the development of driver modules, which provide the test input, call the module or program being tested, and display test output.

There are procedures and constraints associated with each of these methods, although bottom-up testing is often thought to be easier to use. Drivers are often easier to create than stubs, and can serve multiple purposes. Output is also often easier to examine in bottom-up testing, as the output always comes from the module directly above the module under test.

Thread testing

This test technique, which is often used during early integration testing, demonstrates key functional capabilities by testing a string of units that accomplish a specific function in the application. Thread testing and incremental testing are usually utilized together. For example, units can undergo incremental until enough units are integrated and a single business function can be performed, threading through the integrated components.

4.5 Testing Life Cycle

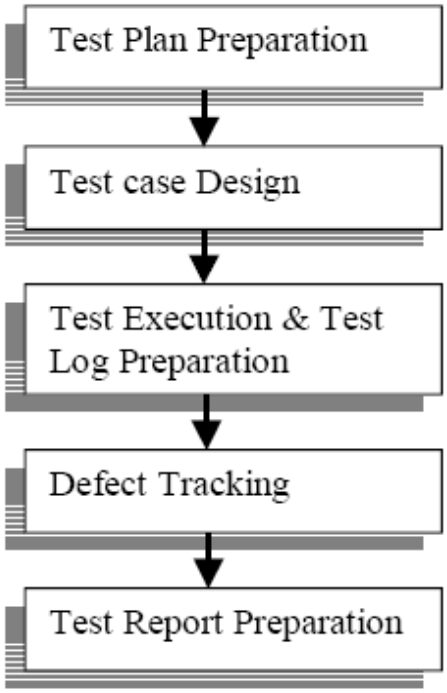

Figure: E Testing life cycle

Test Plan Preparation

The software test plan is the primary means by which software testers communicate to the product development team what they intend to do. The purpose of the software test plan is to prescribe the scope, approach, resource, and schedule of the testing activities. To identify the items being tested, the features to be tested, the testing tasks to be preformed, the personnel responsible for each task, and the risks associated with the plan.

The test plan is simply a by-product of the detailed planning process that's undertaken to create it. It's the planning that matters, not the resulting documents. The ultimate goal of the test planning process is communicating the software test team's intent, its expectations, and its understanding of the testing that's to be performed. The following are the important topics, which helps in preparation of Test plan.

High-Level Expectations

The first topics to address in the planning process are the ones that define the test team's high-level expectations. They are fundamental topics that must be agreed to, by everyone on the project team, but they are often overlooked. They might be considered "too obvious" and assumed to be understood by everyone, but a good tester knows never to assume anything.

People, Places and Things

Test plan needs to identify the people working on the project, what they do, and how to contact them. The test team will likely work with all of them and knowing who they are and how to contact them is very important.

Similarly, where documents are stored, where the software can be downloaded from, where the test tools are located, and so on need to be identified.]

Inter-Group Responsibilities

Inter-Group responsibilities identify tasks and deliverables that potentially affect the test effort. The test team's work is driven by many other functional groups – programmers, project manages, technical writers, and so on. If the responsibilities aren't planned out, the project, specifically the testing, can become a worst or resulting in important tasks been forgotten.

Test phases

To plan the test phases, the test team will look at the proposed development model and decide whether unique phases, or stages, of testing should be performed over the course of the project. The test planning process should identify each proposed test phase and make each phase known to the project team. This process often helps the entire team from and understands the overall development model.

Test strategy

The test strategy describes the approach that the test team will use to test the software both overall and in each phase. Deciding on the strategy is a complex taskone that needs to be made by very experienced testers because it can determine the successes or failure of the test effort.

Bug Reporting

Exactly what process will be used to manage the bugs needs to be planned so that each and every bug is tracked, from when it's found to when it's fixed – and never, ever forgotten.

Metrics and Statistics

Metrics and statistics are the means by which the progress and the success of the project, and the testing, are tracked. The test planning process should identify exactly what information will be gathered, what decisions will be made with them, and who will be responsible for collecting them.

Risks and Issues

A common and very useful part of test planning is to identify potential problem or risky areas of the project – ones that could have an impact on the test effort.

Test Case Design

The test case design specification refines the test approach and identifies the features to be covered by the design and its associated tests. It also identifies the test cases and test procedures, if any, required to accomplish the testing and specifics the feature pass or fail criteria. The purpose of the test design specification is to organize and describe the testing needs to be performed on a specific feature.

The following topics address this purpose and should be part of the test design specification that is created:

Test case ID or identification

A unique identifier that can be used to reference and locate the test design specification the specification should also reference the overall test plan and contain pointers to any other plans or specifications that it references.

Test Case Description

It is a description of the software feature covered by the test design specification for example, "the addition function of calculator," "font size selection and display in word pad," and "video card configuration testing of quick time."

Test case procedure

It is a description of the general approach that will be used to test the features. It should expand on the approach, if any, listed in the test plan, describe the technique to be used, and explain how the results will be verified.

Test case Input or Test Data

It is the input the data to be tested using the test case. The input may be in any form. Different inputs can be tried for the same test case and test the data entered is correct or not.

Expected result

It describes exactly what constitutes a pass and a fail of the tested feature. Which is expected to get from the given input.

Test Execution and Test Log Preparation

After test case design, each and every test case is checked and actual result obtained. After getting actual result, with the expected column in the design stage is compared, if both the actual and expected are same, then the test is passed otherwise it will be treated as failed.

Now the test log is prepared, which consists of entire data that were recorded, whether the test failed or passed. It records each and every test case so that it will be useful at the time of revision.

Example

Test case ID	**Test case Description**	**Test status/ result**
Sys_xyz_01	Checking the login window	Fail
Sys_xyz_02	Checking the main window	True

4.6 Defect Tracking

A defect can be defined in one or two ways. From the producer's viewpoint, a defect is a deviation from specifications, whether missing, wrong, etc. From the Customer's viewpoint, a defect is any that causes customer dissatisfaction, whether in the requirements or not, this is known as "fit for use". It is critical that defects identified at each stage of the project life cycle be tracked to resolution.

Defects are recorded for following major purposes:

- To correct the defect

- To report status of the application
- To gather statistics used to develop defect expectations in future applications
- To improve the software development process

Most project teams utilize some type of tool to support the defect tracking process. This tool could be as simple as a white board or a table created and maintained in a word processor or one of the more robust tools available today, on the market, such as Mercury's Test Director etc. Tools marketed for this purpose usually come with some number of customizable fields for tracking project specific data in addition to the basics. They also provide advanced features such as standard and ad-hoc reporting, e-mail notification to developers and/or testers when a problem is assigned to them, and graphing capabilities.

At a minimum, the tool selected should support the recording and communication significant information about a defect. For example, a defect log could include:

- Defect ID number
- Descriptive defect name and type
- Source of defect -test case or other source
- Defect severity
- Defect priority
- Defect status (e.g. open, fixed, closed, user error, design, and so on) -more Robust tools provide a status history for the defect.
- Date and time tracking for either the most recent status change, or for each
- change in the status history
- Detailed description, including the steps necessary to reproduce the defect
- Component or program where defect was found
- Screen prints, logs, etc. that will aid the developer in resolution process
- Stage of origination
- Person assigned to research and/or correct the defect

Severity versus Priority

The test team based on predefined severity descriptions should assign the severity of a defect objectively. For example a "severity one" defects maybe defined as one that causes data corruption, a system crash, security violations, etc. In large project, it may also be necessary to assign a priority to the defect, which determines the order in which defects should be fixed. The priority assigned to a defect is usually more subjective based upon input from users regarding which defects are most important to them, and therefore should be fixed first.

It is recommended that severity levels be defined at the start of the project so that they intently assigned and understood by the team. This foresight can help test teams avoid the common disagreements with development teams about the criticality of a defect.

Some general principles

- The primary goal is to prevent defects. Wherever this is not possible or practical, the goals are to both find the defect as quickly as possible and minimize the impact of the defect.
- The defect management process, like the entire software development process, should be risk driven, i.e., strategies, priorities and resources should be based on an assessment of the risk and the degree to which the expected impact of risk can be reduced.
- Defect measurement should be integrated into the development process and be used by the project team to improve the development process. In other words, information on defects should be captured at the source as a natural by-product of doing the job. People unrelated to the project or system should not do it.
- As much as possible, the capture and analysis of the information should be automated. There should be a document, which includes a list of tools, which have defect management capabilities and can be used to automate some of the defect management processes.
- Defect information should be used to improve the process. This, in fact, is the primary reason for gathering defect information.

- Imperfect or flawed processes cause most defects. Thus, to prevent defects, the process must be altered.

The Defect Management Process

The key elements of a defect management process are as follows.

- Defect prevention
- Deliverable base-lining
- Defect discovery/defect naming
- Defect resolution
- Process improvement
- Management reporting

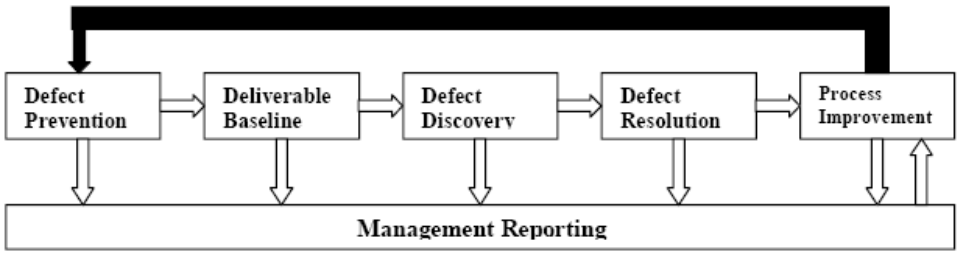

Figure: F Management Reporting

4.7 Test Reports

A final test report should be prepared at the conclusion of each test activity. This might include

- Individual Project Test Report (e.g., a single software system)
- Integration Test Report
- System Test Report
- Acceptance Test Report

The test reports are designed to document the results of testing as defined in the test plan. Without a well-developed test plan, which has been executed in accordance with its criteria, it is difficult to develop a meaningful test report.

It is designed to accomplish three objectives:

- Define the scope of testing - normally a brief recap of the test plan;
- Present the results of testing; and
- Draw conclusions and make recommendations based on those results

The test report may be a combination of electronic data and hard copy. For example, if the function test matrix is maintained electronically, there is no reason to print that, as the paper report will summarize that data, draws the appropriate conclusions, and present recommendations.

The test report has one immediate and three long-term purposes. The immediate purpose is to provide information to the customers of the software system so that they can determine whether the system is ready for production: and if so, to assess the potential consequences and initiate appropriate actions to minimize those consequences.

The first of the three long-term uses is for the project to trace problems in the event the application malfunctions in production. Knowing which functions have been correctly tested and which ones still contain defects can assist in taking corrective action.

The second long-term purpose is to use the data to analyze the rework process for making changes to prevent defects from occurring in the future. Accumulating the results of many test reports to identify which components of the rework process are detect-prone does this. These defect-prone components identify tasks/steps that, if improved, could eliminate or minimize the occurrence of high-frequency defects.

The third long-term purpose is to show what was accomplished.

Individual Project Test Report

These reports focus on individual projects (e.g., software system). When different testers test individual projects, they should prepare a report on their results.

Integration Test Report

Integration testing tests the interfaces between individual projects. A good test plan will identify the interfaces and institute test conditions that will validate

interfaces. Given this, the interface report follows the same format as the individual Project Test report, except that the conditions tested are the interfaces.

System Test Report

A system test plan standard that identified the objectives of testing, what was to be tested, how it was to be tested and when tests should occur. The System Test report should present the results of executing that test plan. If this is maintained electronically, it need only be referenced, not included in the report.

Acceptance Test Report

There are two primary objectives for testing. The first is to ensure that the system as implemented meets the real operating needs of the user or customer. If the defined requirements are those true needs, the testing should have accomplished this objective. The second objective is to ensure that the software system can operate in the real-world user environment, which includes people skills and attitudes, time pressures, changing business conditions, and so forth.

Eight Interim Reports:

1. Functional Testing Status
2. Functions Working Timeline
3. Expected verses Actual Defects Detected Timeline
4. Defects Detected verses Corrected Gap Timeline
5. Average Age of Detected Defects by Type
6. Defect Distribution
7. Relative Defect Distribution
8. Testing Action

Functional Testing Status Report

This report will show percentages of the functions, which have been:

- Fully Tested
- Tested With Open Defects
- Not Tested

Functions Working Timeline report

This report will show the actual plan to have all functions working verses the current status of functions working. An ideal format could be a line graph.

Expected verses Actual Defects Detected report

This report will provide an analysis between the number of defects being generated against the expected number of defects expected from the planning stage

Defects Detected verses corrected Gap report

This report, ideally in a line graph format, will show the number of defects uncovered verses the number of defects being corrected and accepted by the testing group. If the gap grows too large, the project may not be ready when originally planned.

Average Age Detected Defects by Type report

This report will show the average outstanding defects by type (severity 1, severity 2, etc.). In the planning stage, it is benefic determine the acceptable open days by defect type.

Defect Distribution report

This report will show the defect distribution by function or module. It can also include items such as numbers of tests completed.

Relative Defect Distribution report

This report will take the previous report (Defect Distribution) and normalize the level of defects. An example would be one application might be more in depth than another, and would probably have a higher level of defects. However, when normalized over the number of functions or lines of code, would show a more accurate level of defects.

Testing action report

This report can show many different things, including possible shortfalls in testing. Examples of data to show might be number of severity defects, tests that are behind schedule, and other information that would present an accurate testing picture

4.8 Other Testing Terms

Usability Testing

Determines how well the user will be able to understand and interact with the system. It identifies areas of poor human factors design that may make the system

difficult to use. Ideally this test is conducted on a system prototype before development actually beings. If a navigational or operational prototype is not available, screen prints of all of the applications screens or windows can be used to walk the user through various business scenarios.

Conversion Testing

Specifically designed to validate the effectiveness of the conversion process Developers and testers may conduct this test jointly during integration testing, or at the start of system testing, since system testing must be conducted with the converted data. Field -to -Field mapping and data translation is validated and, if a foil copy of production data will be used in the test.

Vendor Validation Testing

Verifies that the functionality of contracted or third party software meets the organization's requirements, prior to accepting it and installing it into a production environment The software vendor and the test team, and focuses on ensuring that all requested functionality has been delivered can conduct this test jointly.

Stress / Load Testing

Conducted to validate the application, database, and network, they may handle projected volumes of users and data effectively. Developers, testers, conduct the test jointly and DBA's and network associates after the system testing. During the test, the complete system is subjected to environmental conditions that defer expectations to answer question such as:

- How large can the database grow before performance degrades?
- At what point will more storage space be required?
- How many users can use the system simultaneously before it slows down or fails?

Performance Testing

Usually conducted in parallel with stress and load testing in order to measure performance against specified service-level objectives under various conditions For instance, one may need to ensure that batch processing will complete within the allocated amount of time, or that on-line response times meet performance requirements.

Recovery Testing

Evaluates the contingency features built into the application for handling inter and for returning to specific points in the application processing. Any restoration, and restart capabilities are also tested here. The test team may conduct this test during system test or by another team specifically gathered for this purpose.

Configuration Testing

In the IT Industry, a large percentage of new applications are either client/server or web-based, validating that they will run on the various combinations of hardware and software. For instance, configuration testing for an web-based application would incorporate versions and releases of operating systems, internet browsers, modem speeds, and various off the shelf applications that might be integrated (e.g. e-mail application)

Benefits Realization Test

With the increased focus on the value of business returns obtained from investments information technology this type of test or analysis is becoming more critical. The benefits Realization Test is a test or analysis conducted after an application is moved into production in order to determine whether the application is likely to deliver the original projected benefits. The analysis is usually conducted by- the business user or client group who requested the project, and results are reported back to executive management.

4.9 Test Standards

External Standards - Familiarity with and adoption of industry test standards from Organizations.

Internal Standards - Development and enforcement of the test standards that testers must meet.

IEEE

- Institute of Electrical and Electronics Engineers
- Founded in 1884
- Have an entire set of standards devoted to Software
- Testers should be familiar with all the standards mentioned in IEEE.

IEEE STANDARDS: That a Tester should be aware of

1. 610.12-1990 IEEE Standard Glossary of Software Engineering Terminology
2. 730-1998 IEEE Standard for Software Quality Assurance Plans
3. 828-1998 IEEE Standard for Software Configuration Management Plan
4. 829-1998 IEEE Standard for Software Test Documentation.
5. 830-1998 IEEE Recommended Practice for Software Requirement Specification
6. 1008-1987 (R1993) IEEE Standard for Software Unit Testing (ANSI)
7. 1012-1998 IEEE Standard for Software Verification and Validation.
8. 1012a-1998 IEEE Standard for Software Verification and Validation Supplement to 1012-1998 Content Map to IEEE 122207.1
9. 1016-1998 IEEE Recommended Practice for Software Descriptions
10. 1028-1997 IEEE Standard for Software Reviews
11. 1044-1993 IEEE Standard classification for Software Anomalies
12. 1045-1992 IEEE Standard for Software Productivity Metrics (ANSI)
13. 1058-1998 IEEE Standard for Software Project Management Plans
14. 1058.1-1987 IEEE Standard for Software Management
15. 1061-1998.1 IEEE Standard for Software Quality Metrics Methodology.

Other Standards:

- **ISO** - International Organization for Standards
- **SPICE** - Software Process Improvement and Capability Determination
- **NIST** - National Institute of Standards and Technology
- **DoD** - Department of Defense

Internal Standards

The use of Standards...

- Simplifies communication
- Promotes consistency and uniformity
- Eliminates the need to invent yet another solution to the same problem
- Provides continuity
- Presents a way of preserving proven practices
- Supplies benchmarks and framework

Chapter 5

5. DSpace DL Software

DSpace is digital library software that allows users to submit, store, and allow others to read and use information that may have broad appeal. With a specific focus on the preservation of stored data, DSpace employs digital preservation functions such as the storage of checksums along with digital objects in order to keep track of and verify a file's conformance with the original. DSpace is an open source software project and is entirely written in Java. DSpace was developed, so that most functionality required by organizations seeking to use such a software product would be covered, in a simple and basic way. DSpace has developed a model that allows users to use the system, submit and use content, and administrators can organize and configure the system. In order to be more usable to different types of users, the software provides a configurable submission and workflow process that can be fit to any organization's information needs.

5.1. DSpace Object Model

While the functionalities of any software is important regarding its use, to have full understanding of a software system it is necessary to look beyond the appearance of that software and get a glimpse at what happens at the backend. DSpace is no different. From simple browsing, DSpace repository user can get a feel of the structure of a DL, while the exact organization and division of its underlying object model remains relatively hidden. DSpace administrators, on the other hand, get a taste of DSpace's underlying structure.

DSpace stores digital content, often referred to as digital objects; thus a very important part of the overall object model of DSpace systems is those objects that are called "Items". Items are organized in a hierarchy in which similar items are

grouped and submitted into Collections of similar content. The highest level of content organization in the system is Communities, which are groups of Collections. As such, a Collection can be in more than one Community. Each Item stored in a DSpace repository is made up of a bundle of bitstreams, so as many files can be stored in a single digital object as needed. Bitstreams adhere to the Bitstream Formats that the system knows about, and DSpace behaves in different ways with different types of objects—e.g., images may have their thumbnails displayed while browsing the system. The DSpace object model diagram is provided in the Figure:G

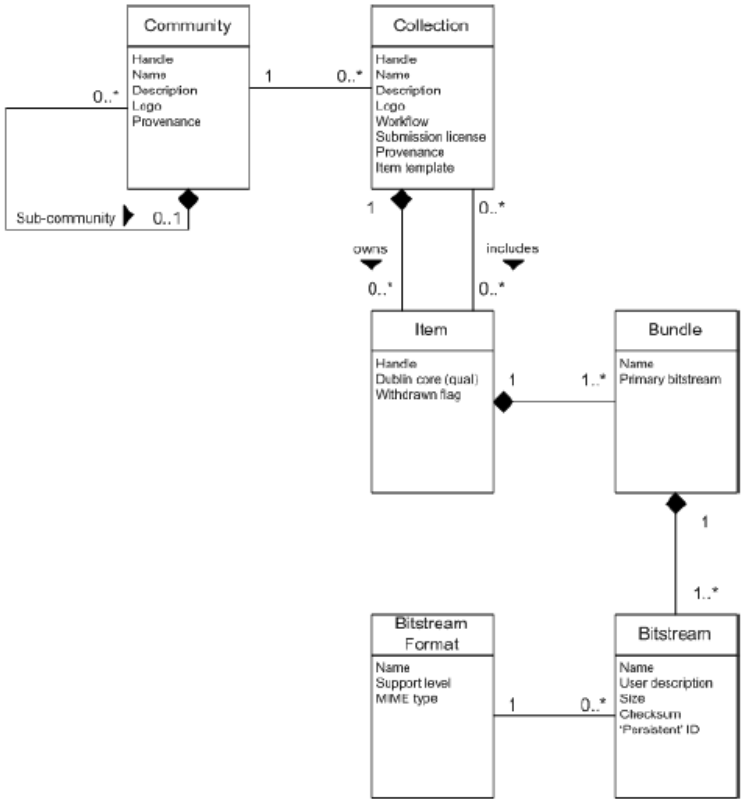

Figure: G DSpace object model

5.2. DSpace Architecture

As detailed in the given Figure: H the DSpace software is divided into a relatively common three-tiered architecture. These three layers are the Application layer,

Business Logic layer, and Storage layer. On the lowest layer of this architecture, the Storage layer, all bitstreams are stored in the repository are as files on the system's file structure. References to these files and most other metadata, settings, and other information that drives the behavior of the system are stored in a relational database system, PostgreSQL.

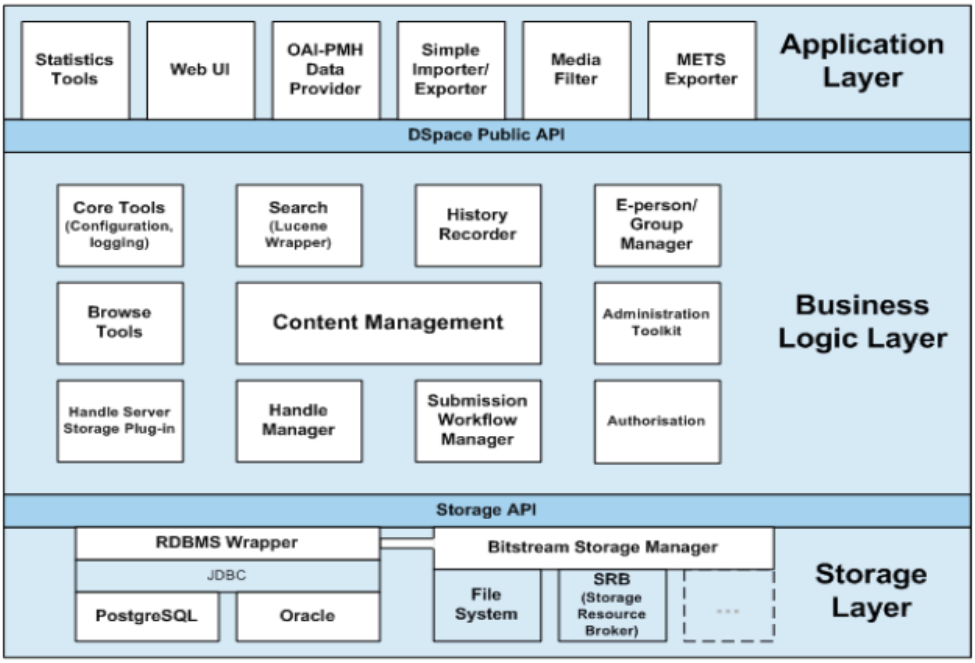

Figure: H DSpace architecture

Together the Storage layer aspects of DSpace make up the Storage API. The Business Logic layer is made up of a set of classes or modules that embody the inner workings of many DSpace object types, including user related functions, browsing and searching related aspects, content management, and others. Business Logic classes make up DSpace's Public API, which allows third party code to interact with DSpace in the same way that typical interaction within the software occurs. Lastly, the Application layer is the highest level layer of functionality in DSpace and brings together DSpace backend functionalities to provide the services and functionality that users see when they use the system. Included in the Application layer are the import/export functionality the software

provides, statistics tools, and the web-based user interface. Given DSpace's open source nature, all of these software aspects have source code available on Internet for download for organizations to tweak and customize DSpace as per their needs.

5.3.DSpace Ingestion

DSpace is a software that serves as a repository and stores digital content. In a system with such a goal, perhaps the most critical aspect of the system is how that data enters the system. This occurs mainly in two ways in DSpace. The web based UI for the software allows a user to submit items to collections as long as they are logged in as a registered user. When users log in, they go through a configurable workflow where they upload and describe their submissions. (Workflows in DSpace are discussed further in the next section)

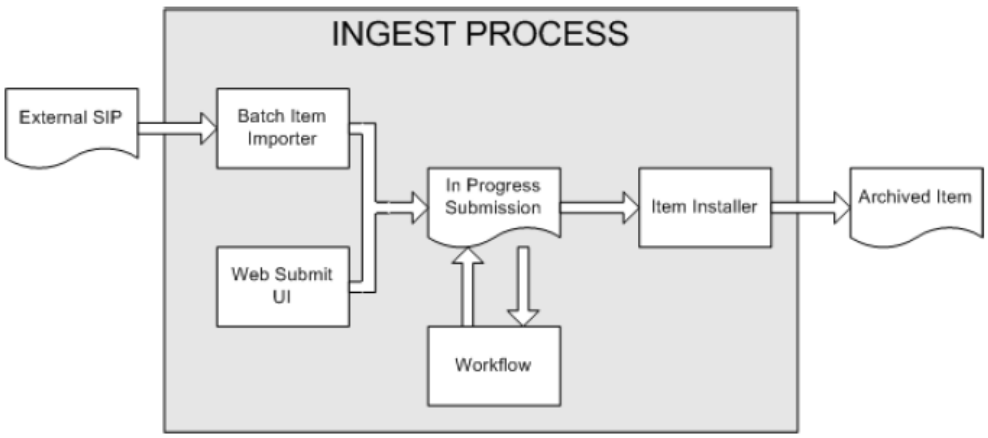

Figure: I DSpace Ingest Process

Alternatively, DSpace administrators who have a large amount of content to be batch imported may take advantage of the import/export functionalities of the system. The Item Importer is a command line tool that comes bundled with the system and allows users to import collections of content into the system.

```
archive_directory/
    item_000/
        dublin_core.xml -- qualified Dublin Core metadata
        contents        -- text file containing one line per filename
        file_1.doc      -- files to be added as bitstreams to the item
        file_2.pdf
    item_001/
        dublin_core.xml
        contents
        file_1.png
        ...
```

Figure: J DSpace's simple archive format for importing and exporting

The Item Importer uses DSpace's simple archive format, which is a simple directory structure that holds items for import into the system. (An example of a simple archive is given above in Figure: J). A top level archive directory contains uniquely named directory, each of which contains everything necessary to import a single item. Each sub-folder is required to contain two files, in addition to the actual content to be imported. The required file "dublin_core.xml" contains an XML representation of qualified Dublin Core element names and the textual content that contains metadata records, including author, title, and so on. A plain text "content" file has one line containing the filename of each file that will be included in that digital object. Once this structure is put in place, the import command can simply be run and all content will be imported into the repository. The tool provides a "map file" after being run, which details all items that were imported and their new location within the system—this file can be helpful in future for exports or removal of imported contents.

5.4. DSpace Workflow

The DSpace submission workflow system is a critical part of the DSpace architecture that allows submission, processing, and final addition of content to the live repository. DSpace's underlying model includes E-People, users who have registered with the system and have certain authorizations, roles, rights, and privileges that translate abilities to complete certain tasks within the DSpace system. A typical submission begins with the system asking the user a couple of

questions about digital document to be added in the repository and number of files involved in the submission. Then the system guides the user through the different steps, which are outlined in the following Figure: K.

Workflow Step	Description
1. Describe	User enters metadata about the document(s) they are submitting, including but not limited to authors, title, keywords, and a description.
2. Upload	The user selects and uploads the files on their local machine that they'd like to upload as part of the submission. Each file's type is identified by the system and the user verifies the file type.
3. Verify	An overview of all details of the submission are given including a summary of the entered metadata and the files involved in the submission.
4. License	The user is shown and must agree to the license the system administrator has assigned to submitted content for this collection.
5. Complete	The user's actions in the submission process are complete. Based on the workflow steps set for the collection, the item may immediately be added to the collection or have to be reviewed by system administrators before its addition to the collection.

Figure: K DSpace submission workflow overview

Chapter 6

6. DSpace Functional Testing using Test Director

In this project functionality testing of DSpace digital library software has been done using the automated software testing tool named as "TestDirector of Mercury Interactive (Proprietary)". TestDirector helps in organizing and managing all phases of the software testing process, including planning, creating tests, executing tests, and tracking defects.

Using TestDirector, we can start up a testing project and maintain a database of test cases which can be used to build *test sets* (groups of tests) executed to achieve some specific goal. For example, we can create a test set that checks a new version of the software, or one that checks a specific feature of the software. As we execute tests, one of the important aspect of TestDirector is it lets us report defects detected in the software under test. Defect records are stored in a database where we can track them until they are resolved in the software.

Mercury TestDirector allows deploying high-quality applications quickly and effectively by providing a consistent, repeatable process for requirements gathering, planning and scheduling tests, analyzing results, and managing defects and issues. TestDirector is a single, application for all essential aspects of test management — Requirements Management, Test Plan, Test Lab, and Defects Management.

DSpace is an open source fully developed software; it is being widely used nationally and internationally for creating and maintaining digitally repository. For better understanding of the software and accelerating its usage it is very much necessary to conduct its testing as far as its functionality and performance is concerned.

For test purpose the below mentioned DSpace release version and other softwares have been used:

DSpace – Release 1.4.2 (Installed on Linux Fedora Core 6)

Apache-tomcat-6.0.10

PostgreSQL 8.1.4

ANT 1.6.5

Sun java - jdk1.5.0_02

Data inputs for the testing

Testing project in TestDirector can be completed in three phases:

1. Plan Phase – Test cases preparation
2. Run Phase – Run the test cases grouped in test sets
3. Defect Phase – Track the defects

Plan Phase

To perform the DSpace functionality test using TestDirector, a project database has been set up using the Project Administration Utility of TestDirector. In the database various test cases has been prepared based on the functionality of the DSpace software. In each test case steps to be followed is defined to conduct the respective function in the software.

Run Phase

In this phase the test cases are grouped into test sets to run the tests and observe the result.

Defect Phase

In this phase defects can be tracked.

The followings are the some of the functional features of DSpace on which testing has been conducted. The complete "*Test Report Documentation*" generated by the

tool at the completion of the project is attached as an Appendix in the project report.

6.1 CORE CONCEPTS AND RELATIONSHIPS

The following are the core concepts of DSpace and their relationships which helps users in better understanding the DSpace.

6.1.1 Site

A DSpace Site is a web site which deals with the details of the host institution where DSpace is installed and various services are offered
(See the appendix for the test result)

6.1.2 User

A DSpace User is an individual who uses the DSpace system, by visiting a DSpace site with a web browser. DSpace users can be at any given time either unknown, or credentialed to the system to some degree, for example via username/password, LDAP server authentication or IP-address-based network presence. DSpace keeps some basic information of *registered users* (email address, name, and credential information), so that they can take advantage of all of the systems functionality (For example: Submission, My DSpace).
(See the appendix for the test result)

6.1.3 Group of Users

In DSpace, repository administrators can organize DSpace users into groups, which may be used to define participants in a role within a collection's submission process (e.g. "approver"). Policy statements can also refer to groups of users (for example, allow users in group "thesis-administrators" to edit the collection metadata for the thesis collection).
(See the appendix for the test result)

6.1.4 Community

A DSpace Community is a convenient entry point or "portal" into the corpus of material in the repository. A Community consists a home page, which can be edited, a set of collections referred to by the community, and a group of users with management and administrative responsibility for the community.
(See the appendix for the test result)

6.1.5 Collection

A DSpace Collection is a group or a set of DSpace items that are related in some way. A DSpace Collection consists of a configurable home page for the collection, a set of items referred to by the collection, a configurable submission process for content entry into the collection, and a group of users with management and administrative responsibility for the collection.

Users who submit items to DSpace can choose a collection to submit to. Further, DSpace administrators can re-organize items into another collection or even multiple collections after their initial submission. Collections typically contain items that are similar in some dimension (for example: source, purpose, existing series or audience, subject matter, research topic). Administrators can also use collections to organize a submission process for consistency of submitted content (for example, with respect to scope of content, metadata requirements, required bitstream formats, etc.)
(See the appendix for the test result)

6.1.6 Approval Process

In institutional settings, decisions about what enters into the repository can be set through approval process. In DSpace, each collection can have its own approval process, and specify individuals who will participate in it. Such approval

processes can range from very simple, to multistage review with multiple individuals participating at each stage.
(See the appendix for the test result)

6.1.7 Item

A DSpace Item is a logical grouping of a useful set of content and metadata that are related in some way. Items correspond to "Archival Bitsreams" in DSpace.
Examples of DSpace items include:
A working paper, a conference presentation, a monograph (book), an annotated series of images, a video clip, materials for a course lecture, a research paper with auxiliary material (e.g. dataset, extended bibliography, rich media images, PDF, doc files etc.).
(See the appendix for the test result)

6.2 SUBMIT

In DSpace registered users can submit their items by selecting the appropriate communities and collection for which they are authorized to or have access.

6.2.1 Anonymous *vs.* Credential Access

Any user can use DSpace to search and browse items and collections that are globally accessible. Only users who are registered with DSpace can submit items, administer items, collections, or communities, or view items that are not globally accessible. DSpace software will ask user to authenticate themselves to the system whenever the system's current policy configuration indicates that some credential is required to perform the action.
(See the appendix for the test result)

6.2.2 Register with DSpace

In DSpace users can be registered through the networked systems or LDAP server as well as the user can registered themselves through the following DSpace registration process:

1. DSpace gathers the user's email address.
2. Verifies that individuals are registering using an email address that they can access. In other words, it is not possible to register for DSpace using someone else's email address.
3. Allows the user to initially set their user profile: password and basic personal information (name, etc.).
4. Store the password securely within the system

(See the appendix for the test result)

6.2.3 Forgotten Password

DSpace provides a secure process for users who have forgotten their password to select a new password without human administrative intervention. Users provide their email address, and the system mails them a special hyperlink which, when clicked, allows them to update their password.

(See the appendix for the test result)

6.2.4 Edit user Profile

DSpace enables registered users to edit the basic personal information that the system keeps for them such as:

- Lastname, Firstname
- Contact Telephone (optional)
- Password

The system keeps track of users by the single email address with which they initially registered. A user's mail address cannot currently be changed via the end-

user interface, but can be changed using the administrative user interface. Thus, end users can get their email updated by contacting administrator
(See the appendix for the test result)

6.2.5 Submit to Collection

DSpace provides an easy way for materials to enter into the archive in a distributed fashion across the host institution. The submitted materials gets enter into the DSpace archive through the submission process. Materials within DSpace are always in one of three stages:

1. Items Being Assembled – the submitting user is still in the process of entering the Item's Metadata, or uploading the file(s) to be included with the item.
2. Submissions Pending Archive – the submitting user has submitted the item to the initial Collection's submission approval process. But this approval process is not yet complete, and the item is not yet archived as it requires more authentications.
3. Archived Items – the item is archived, having been approved for entry into the collection to which it was submitted.

Authorized users can initiate submissions to a DSpace collection that they have chosen. Users can choose the collection to submit to in one of two ways:

1. Explicitly, from a list of all collections in DSpace (accessible from DSpace home)
2. Contextually, by first navigating to the collection of interest (accessible from each collection's home page)

(See the appendix for the test result)

6.2.6 Submit Baseline Metadata

In DSpace, users can specify baseline metadata with their submission. Users fill in the form for all submitted items. This baseline metadata will enable users to search across all of DSpace, and to easily find submitted items in the future. The baseline metadata requested for each submitted item is based upon the qualified Dublin Core Metadata Schema, adapted to DSpace requirements.

The elements captured through the end-user interface are:

1. Author(s);

Zero or more supported. Currently no authority control for authors (i.e. DSpace does not currently know that “Samuel Clemens” and “Mark Twain” are the same author, nor does it distinguish well between two authors that share the same name).

2. Title(s);

Including alternative titles, if applicable.

3. Date of Issue

4. Series Name and Report Number; For example “Sloan School of Management Working Papers, Number 2002-128” If applicable. Zero or more supported.

5. Identifiers;

Including ISBN, ISSN, ISMN, URI, and other. Zero or more supported.

6. Language (in which submitted material is written).

7. Subject Keywords;

Zero or more supported. Currently no thesauri or authority control for subject keywords.

8. Abstract

9. Sponsors / Funding Codes

10. Other Description

Additional elements are modeled and stored internally. Some of these are generated automatically by the system. Others may be managed by DSpace administrators using the admin user interface.

(See the appendix for the test result)

6.2.7 Submit Domain-Specific Metadata

In addition to DSpace baseline metadata, Users can submit domain-specific metadata that is specific to their item, or that is required by the curators of the collection to which they are submitting, by uploading a file that contains the relevant metadata. For example, a collection of images might require that each image be submitted with GIS metadata indicating the location of the corresponding image. (write more explicitly here about example you are quoting)

Users may annotate their item using as many metadata formats as desired. DSpace calculates and maintains an MD5 checksum of each file uploaded with the item that can be used by DSpace administrators and by users to verify the integrity of the content and metadata within the system. DSpace maintains each of these metadata bitstreams with the item, and make them available with the item.

(See the appendix for the test result)

6.2.8 Upload File(s) to Item

Submitted DSpace Items are supposed to become “archival Bitsreams”, that is actual digital documents or files . As such, each item can include multiple pieces of content. Each piece of content might comprise several files.

Users must submit one or more files to be included with their item. For example, a user might submit a conference paper, along with presentation materials actually used at the conference. A researcher might submit a pre-print of an article, along

with dataset(s) that would enable other researchers to independently reproduce the research results.

DSpace calculates and retains a checksum of each file uploaded with the item that can be used by DSpace administrators and by users to verify the integrity of the content and metadata within the system. This checksum can be obtained through the end-user interface.
DSpace attempts to recognize the bitstream format of each uploaded file. If it cannot match the bitstream format to a known format from the system's bitstream format registry, it asks the user to describe the format so that library administrators can track and support important emerging formats over time.
(See the appendix for the test result)

6.2.9 Grant Distribution License

To enable the host institution to administer, preserve, and distribute the submitted material, DSpace asks the user to grant the host institution a non-exclusive license to distribute the material, and to translate it for the purposes of preservation.

License agreements can vary by Collection, and are specified by the collection administrator(s) for the Collection. Because license terms are likely to change over time based on the needs of submitters and the host institution, DSpace stores a copy of the license that was granted at the time of submission as a bitstream within the item, so that the specific terms agreed upon are always available.
(See the appendix for the test result)

6.2.10 Augment & Approve Submissions

When the user submits an item to a DSpace collection, the system routes the submission through the approval process previously configured for that collection. The approval process can vary by collection, and can include any combination of

the approval roles defined below. Further, groups of users can be used to associate any number of users with each of the roles that are configured for the process. The system routes the submission to individuals (if any) who have been chosen to assume each of the following roles for the target collection:

Reviewers: review the content of the submission for appropriateness to the collection. Reviewers act as the gatekeepers for the collection. Reviewers are empowered to return a submission back to the submitter because it is deemed inappropriate for the collection. Reviewers do not edit submission metadata.

Approvers: check the submission for completeness and/or obvious errors (e.g. wrong file uploaded). Approvers can edit the submission's metadata to fix obvious errors, and are empowered to return a submission back to the submitter because it is incomplete or in error.

Metadata Editors: check and/or augment the submission's metadata. For example, a metadata editor may be assigned to add the appropriate series name and number to each submission in a collection. Metadata Editors can only edit the submission's metadata.

Each collection may specify zero or more individuals to assume each of these roles. If zero individuals are chosen, the system skips that stage of the approval process. For example, a collection might have a simple process for a series of working papers where all submissions are routed to a single staff metadata editor. This metadata editor would assign the series number to each submission. All submitted items would be approved and enter the archive – in this configuration it would be the responsibility of each submitter to crosscheck their own submissions and decide that they were suitable for the archive. Another collection might additionally define a reviewer role to ensure that submitted material is of appropriate nature to be included in the collection. Yet another collection might use all three roles. The intent is to allow communities flexibility in meeting their

collection management needs while avoiding the institutional paralysis that often results from too much flexibility or too many choices.

DSpace users in each role are notified via email when they have pending submissions that require their attention. These tasks are also available to them from the "My DSpace" section of the DSpace system.
As each task is completed, the submission is routed to the individual(s) responsible for the next stage of review, if any. The system sends email(s) to each of these Users, with links to their pending task in the DSpace system.
When the submission either enters the DSpace archive or is returned to the submitter, the system notifies the submitter by email about the status of their submission.
(See the appendix for the test result)

6.2.11 My DSpace

DSpace offers each user personalized access to information within the system through their My DSpace page. As appropriate given their role(s) in the system, users can view their:

1. Items being assembled.
2. Submissions pending archive.
3. Archived Items that they submitted.
4. Review tasks for Items pending archive.

DSpace filters these sections so that only the sections relevant to each user are presented such as:

1. View Archived Items Submitted by User:
Users can easily view all of the archived items that they submitted from their personalized My DSpace page.

2. View Items being Assembled

Users can access submissions that were interrupted while partially complete (because key information was not available, because all files were not available for upload, or simply because the user was called away for an extended coffee break) from the "Items being Assembled " section of their personalized My DSpace page.

3. View Pending Approval Tasks

Users who are taking part in the approval process for one or more collections within DSpace can view, select, and perform tasks that require their attention from the "Pending Approval Tasks" section of their personalized My DSpace page.

(See the appendix for the test result)

6.3 ARCHIVE

DSpace maintains the archives of the items submitted and are accessible through the web interface. For the archival of items submitted it undergoes through the following process such as:

6.3.1 Assign/Resolve Cietable Persistent Name

DSpace creates a persistent name for submitted material that enters the archive. The name that DSpace assigns is intended to be valid and resolvable in perpetuity; the institution hosting the DSpace system stands behind that commitment, even if the DSpace system in place at the time of submission is removed, revised, or replaced. The original host institution may at some point assign that commitment to another institution (conveying responsibility for maintenance/administration of the submitted material to the other institution). DSpace currently creates and assigns persistent names for archived items. The names assigned are free of semantics about the material that they refer to. Further, they are decoupled from both logical and physical storage currently in use for associated material.

Assigned names are registered using the CNRI Handle System. CNRI provides an http resolver at:
http://hdl.handle.net/<handlename>
CNRI also makes plug-ins available for most popular browsers that allow Handles to be resolved without need for an http resolver. DSpace does not require that DSpace users install the handle plug-in, and is agnostic about its use. The name assigned by DSpace is appropriate for citing the submitted material from other digital materials, or from print.
(See the appendix for the test result)

6.3.2 Store Bitstreams

DSpace provides storage for the individual bitstream(s) associated with submitted, archived, items. Users can access the stored bitstreams for any item through that item's overview. DSpace allows additional physical storage for bitstreams to be added, and for stored bitstreams to be easily moved from one physical store to another.
(See the appendix for the test result)

6.3.3 Generate Bitstream Checksums

For each bitstream maintained within the system, DSpace generates and stores an MD5 checksum that can be used either by users or by downstream preservation services to verify the integrity of the stored bitstreams over time. Checksums for each of an Item's bitstreams are accessible from the Item Overview.
(See the appendix for the test result)

6.3.4 Identify Bitstream Format and Preservation

DSpace maintains a bitstream format for each bitstream stored within the system, in addition to maintaining the sequence of bits associated with that bitstream.

Bitstream formats are chosen from a registry of known formats, the Bitstream Format Registry, which is maintained by DSpace administrators.
DSpace attempts to identify the bitstream format automatically. If it is unable to do so, it requests that the submitting user identify the bitstream format from those within the bitstream format registry.
If the bitstream format of a submitted bitstream has not yet been entered into the bitstream format registry, the system requests a prose description of the format, so that DSpace administrators can consider whether or not this format should be incorporated into the registry.

DSpace administrators can identify a preservation service level for each entry in the bitstream format registry.
(See the appendix for the test result)

6.3.5 Store Item Provenance Information

DSpace maintains provenance information for each item in the archive, in qualified Dublin Core <description.provenance> values. In addition to the system generated provenance values described, administrators can annotate items with additional human generated provenance values.
(See the appendix for the test result)

6.4 DISSEMINATE

The items submitted and archived into the DSpace digital library repository can be disseminated and accessed by the users through search and browse.

6.4.1 Search

DSpace offers users the capability to search DSpace for items of interest both simple and advanced.
(See the appendix for the test result)

6.4.2 Search Results

DSpace performs the search and produces a result set once the user specifies the search terms. DSpace displays the result set, including a terse description for each item in the results. From the terse item description displayed in the search results, the user may select a desired item to view its Item Overview.

(See the appendix for the test result)

6.4.3 Browse Items

From the DSpace home page, users can browse all items in DSpace by title, author, or issue date.

(See the appendix for the test result)

6.4.4 Browse Communities and Collection

DSpace administrators organize DSpace items into collections, and include collections in communities. Users can browse the structure of communities and collections, in an outline view. Each outline entry includes text describing the contents and/or purpose of the corresponding community or collection. Users can select an entry from the outline view to access the home page of the selected community or collection, from which further bounded browse or search can be performed.

(See the appendix for the test result)

6.4.5 Recent Submissions

Recently submitted items to each collection are displayed in a sidebar on each collection’s home page.

(See the appendix for the test result)

6.4.6 View Item Overview

Terse item descriptions in either the search view or browse view are linked to the corresponding item overview. Authorized users can view an item's overview, which displays the item's DSpace core metadata, lists the collection(s) that include it, and provides links to each of the bitstreams that the item comprises.

(See the appendix for the test result)

6.4.7 Download Item Bitstream(s)

Users may download each of the bitstream(s) in the item by selecting the bitstream from those listed in the item's item overview if the user is an authorized user to access bitstream in the repository.

(See the appendix for the test result)

6.5 MANAGE AND ADMINISTER

In DSpace there is a provision of administrator account which can be created through the command mode after the installation of the software. DSpace administrator can perform various administrative tasks through command mode as well as from the administrative user interface. From administrative user interface DSpace administrator can conveniently access the administrative functionality in addition to the end user interface. This section describes the functionality of DSpace administrative tasks.

6.5.1 Administer Communities and Collections

1. Create/Delete Community/Collection

Authorized DSpace users can create a new community, or create a new collection and associate it with an existing community. Admin users can also delete an existing community or collection.

2. Edit Community/Collection Home Page

Authorized users can edit key information about a community or collection. The information is used primarily within the home page of the community or collection, and includes:

- Name
- Short Description (used in community/collection browse)
- Intro Text
- Logo (upload bitstream from local filesystem)
- Copyright Text
- Sidebar Text

Apart from the above there is some Intro Text, Copyright Text, and Sidebar Text may include HTML. This allows some degree of user formatting control, and allows easy links to pages outside DSpace to be easily embedded.
(See the appendix for the test result)

6.5.2 Administer Items

Authorized users in DSpace can change the content and metadata of an item, or expunge the item.

1. Add/Change/Remove Metadata Values

Authorized users can use the admin user interface to:

- Add baseline metadata: that is, add to an item any number of values for any qualifier in the system's registry of Dublin Core Qualifiers.
- Change baseline metadata: that is, edit any existing value(s) for any qualifier
- Remove baseline metadata: that is, remove any/all existing value(s) for any qualifier

2. Add/Delete Bitstreams

Authorized users can use the admin user interface to:

- Add to an item any number of additional bitstreams

- Remove any existing bitstream(s) from the item

3. Expunge Item

- Authorized users can use the admin UI to expunge an item from the system. This deletes the item completely

(See the appendix for the test result)

6.5.3 Administer Users and Groups

1. Users

Authorized users can use the admin user interface to:

- Identify users who may register with DSpace. The email address of valid users must be entered through this way before users will be able to register for the system.
- Edit the personal information stored for a specified user, including their email, first name, last name, phone, password, and whether or not they are active in DSpace.
- Delete a specified user from the system.

2. Groups

Authorized users can create and maintain groups of users. DSpace uses these groups and authorization policies to determine which users are authorized to perform particular actions within the system.

Authorized users can use the admin user interface to:

- Create a group of users
- Name the group
- Add users to the group
- Remove users from the group

(See the appendix for the test result)

6.5.4 Authenticate / Authorize Users

DSpace uses a set of machine-interpretable policy statements to determine whether or not a given user is authorized to perform a given action upon a given resource.

(See the appendix for the test result)

6.5.6 List / Abort Submission Processes

Authorized users can view the state of open submission processes in the system, and can abort a selected submission process. When a submission process is aborted, any pending tasks are removed from the "My DSpace" task and lists of approvers, reviewers, and/or metadata editors. The submitted item remains accessible to the submitter from their "My DSpace" page.

(See the appendix for the test result)

6.5.7 Manage Bitstream Format Registry

Authorized users can add, edit, or delete a file format in the system's registry of bitstream formats. DSpace uses this registry to allow users to identify the format of bitstreams they submit to DSpace.

(See the appendix for the test result)

6.5.8 Bulk Import

DSpace system administrators can bulk import items into the system, and bulk import users to the system. Unlike all of the preceding administrative functionality, the bulk import tools are not available in the web-client-based administrative user interface. Rather, bulk import tools are driven from a command-line interface.

(See the appendix for the test result)

Conclusion

In this project functionality testing of DSpace digital library software has been undertaken and the detailed study of concepts behind software testing, principles and guidelines to handle a software testing projects has been presented. As DSpace is an open source digital library software and it is being widely used to create digital library repository in each sphere of the world from an individual digital repository to an university level and even by the corporate industries to build up and maintain their digital archives. The basic aim of this project is to perform an exhaustive testing of the DSpace which can help in validation and analysis of the software as far as its functionality is concerned.

Testing has been done on various functional features of DSpace following the step by step process using the automated software testing tool and consecutively report has been generated for each test sets as the test result which is attached as *Appendix* in this project. This testing will be helpful in better understanding of the software functionality and can be used as a guide map for practicing DSpace.

While testing, the software was found stable enough, platform independent, flexible, well documented and fully satisfies its specification in order to address the need and requirement of any standard “Digital Library”.

Though the software is mature and being successfully used across the Globe still there is need to add some new features in it like:

- Prioritized List display of Communities and collection (If required)
- Bulk import from GUI (Graphic User Interface)
- Version Control for document management
- Authority Control
- Adding thesaurus

Conclusion

While testing the software it was found that during the following test cases the software status showed as failed

- Bulk Import (DSpace web UI does not have bulk import facility)
- The test result also mentioned that DSpace does not have version control feature.
- Dspace currently does not support Thesauri or authority control for subject keywords.
- DSpace does not support prioritized list display of communities and collection

The aim of the present study is to explore how the software testing is carried out for any software. During the study some of the features of DSpace were tested through the testing software. This shows that it is very essential to carry out software testing before actually using any software especially when it is open source software to understand any software in more detail.

This project will certainly add value for further studies in finding out shortcomings in the software and any enhancement needed in further development of the software taken for testing.

Bibliography

References:

1. Patton, Ron. Software Testing. 2nd ed. USA: SAMS, 2006.
2. Kaner, Cern et.al. Lessons Learned in Software Testing: a context driven approach. New York: Wiley computer publishing, 2001.
3. Perry, Willian E. Effective Methods of Software Testing. 3rd ed. USA: Wiley, 2006.
4. Black, Rex. Pragmatic Software Testing. USA: Wiley, 2007.
5. Jorgensen, Paul C. Software Testing: A creftsman's approach. 2nd ed. Florida: CRC Press, 2002.
6. Graham, Dorothy. Foundation of Software Testing: ISTQB Certification. Int. Thomson Business Press. 2006.
7. J. M. Ockerbloom, Toward the next generation: Recommendations for the next DSpace Architecture, DSpace Architecture Review Group, San Antonio, TX, Report 1/24/2007, 2007.
8. M. Smith, M. Barton, M. Bass, M. Branshofsky, G. McClellan, D. Stuve, R. Tansley, and J.H. Walker, "DSpace – An Open Source Dynamic Digital Repository," in D-Lib Magazine. Vol. 9, 2003.
9. T. Donohue and D. Salo, "DSpace How-To Guide," JCDL 2006, Chapel Hill, NC 6/11/2007.
10. DSpace Open Source Software. 24th June. 2007 http://www.dspace.org/
11. Goh, Dion Hoe-Lian et.al. "A Checklist for evaluating open source digital library software." Emerald Full Text Articles Volume 30 Number 4 2006 pp. 360-379. 2 Aug. 2007. http://www.emeraldinsight.com/Insight/ViewContentServlet?Filename=Published/EmeraldFullTextArticle/Articles/2640300403.html.

Appendix: DSpace Functionality Test

Description: DSpace is an open source digital library software. This test has been undertaken to test the functionality of the Software

Creator: Mohammad Ehtesham
E-mail: ehtesham@drtc.isibang.ac.in
Creation Date: 8/31/2007

Table of Contents

Chapter 1 Tests

Chapter 2 Site

Subject : Core Concepts and Relationships
Status : Ready
Designer : admin
Creation Date : 8/16/2007
Type : MANUAL
Description : A Dspace Site is a specific installation of the DSpace software, upon which Digital Library services are offered that are backed by the host institution. The DSpace Site URL and the Host name has to be defined while installation in the dspace.cfg file within the DSpace directory.

"/dspace/config/dspace.cfg

For e.g:

Basic information

# DSpace installation directory
dspace.dir = /dspace

# DSpace base URL. Include port number etc., but NOT trailing slash
dspace.url = http://localhost/dspace

# DSpace host name - should match base URL. Do not include port number
dspace.hostname = localhost.localdomain

# Name of the site
dspace.name = DSpace @ My University

After installation the installation can be accessed through browser with url:

http://localhost/dspace

Chapter 3 User

Subject : Core Concepts and Relationships
Status : Ready
Designer : admin
Creation Date : 8/16/2007
Type : MANUAL
Description : A DSpace User is an individual who uses the DSpace system, by visiting a DSpace site with their web browser. DSpace users can be at any given time either Anonymous, or Credentialed to the system to some degree, for example via username/password or -address-based network presence. DSpace keeps some basic information for registered users (email

address, name, credential information), so that they can take advantage of all of the systems functionality (For example:
Submission, My DSpace).

Chapter 4 Group of Users

Subject : Core Concepts and Relationships
Status : Ready
Designer : admin
Creation Date : 8/16/2007
Type : MANUAL
Description : DSpace administrators can organize DSpace users into groups, which may be used to define participants in a role within a collection's submission process (e.g. "approver"). Policy statements can also refer to groups of users (for example, allow users in group "thesis-administrators" to edit the collection metadata for the thesis collection).

While it is true that there may be some organizational or sociopolitical group of people that correspond to an overall social "community", DSpace functionality is concerned with defined groups for specific roles within the community.
For example:
Who can edit the community's home page?
Who can add collections to the community?
Who can submit items to a collection's submission process?

Chapter 5 Community

Subject : Core Concepts and Relationships
Status : Ready
Designer : admin
Creation Date : 8/16/2007
Type : MANUAL
Description : A DSpace Community is a convenient entry point or "portal" into the corpus of material in the repository. A Community consists of a configurable home page for the community, a set of collections referred to by the community, and a group of users with management and administrative responsibility for the community. Because communities must be administered, DSpace communities typically correspond – at least initially – to an organizational entity, for example, a school, department, laboratory, or research center.

Chapter 6 Collection

Subject : Core Concepts and Relationships
Status : Ready
Designer : admin
Creation Date : 8/16/2007
Type : MANUAL
Description : A DSpace Collection groups together a set of DSpace items that are related in some way. A DSpace Collection consists of a configurable home page for the collection, a set of

items referred to by the collection, a configurable submission process for content entry into the collection, and a group of users with management and administrative responsibility for the collection. Users who submit items to DSpace can choose a collection to submit to. Further, DSpace administrators can re-organize items into another collection – or even multiple collections – after their initial submission. Collections typically contain items that are similar in some dimension (for example: source, purpose, existing series or audience, subject matter, research topic). Administrators can also use collections to organize a submission process for consistency of submitted content (for example, with respect to scope of content, metadata requirements, required bitstream formats, etc.)

Chapter 7 Approval Process

Subject : Core Concepts and Relationships
Status : Ready
Designer : admin
Creation Date : 8/16/2007
Type : MANUAL
Description : In institutional settings, decisions about what enters the archive often must be distributed, with decisions made close to communities. In DSpace, each collection can have its own approval process, and specify individuals who will participate in it. Such approval processes can range from very simple, to multistage review with multiple individuals participating at each stage.

Chapter 8 Item

Subject : Core Concepts and Relationships
Status : Ready
Designer : admin
Creation Date : 8/16/2007
Type : MANUAL
Description : A DSpace Item is a logical grouping of a useful set of content and metadata that are related in some way. Items correspond to “Archival Atoms” in DSpace. Examples of DSpace items include:

A working paper, a conference presentation, a monograph (book), an annotated series of images, a video clip, materials for a course lecture, a research paper with auxiliary material (e.g. dataset, extended bibliography, rich media images).

Chapter 9 Anonymous vs Credential Access

Subject : Submit
Status : Ready
Designer : admin
Creation Date : 8/16/2007
Type : MANUAL

Description : Any user can use DSpace to search and browse items and collections that are globally accessible. Only users who are registered with DSpace can submit items, administer items, collections, or communities, or view items that are not globally accessible.

DSpace will ask that the user authenticate themselves to the system whenever the system's current policy configuration indicates that some credential is required to perform the action.

Chapter 10 Register with DSpace

Subject : Submit
Status : Ready
Designer : admin
Creation Date : 8/16/2007
Type : MANUAL
Description : The user registration process:

1· Gathers the user's email address
2· Verifies that individuals are registering using an email address that they can access. In other words, it is not possible to register for DSpace using someone else's email address.
3· Allows the user to initially set their user profile: password and basic personal information (name, etc.).
4· Stores the password securely within the system

Chapter 11 Forgotten Password

Subject : Submit
Status : Ready
Designer : admin
Creation Date : 8/16/2007
Type : MANUAL
Description : DSpace provides a secure process for users who have forgotten their password to select a new password without human administrative intervention. Users provide their email address, and the system mails them a special hyperlink which, when clicked, allows them to update their password.

Chapter 12 Edit user Profile

Subject : Submit
Status : Ready
Designer : admin
Creation Date : 8/16/2007
Type : MANUAL
Description : Registered DSpace users can edit the basic personal information that the system keeps for them:
· Lastname, Firstname
· Contact Telephone (optional)
· Password

The system tracks users by the single email address with which they initially registered. A user's mail address cannot currently be changed via the end-user interface, but can be changed using the administrative user interface. Thus, end users can update their email address through a call to the help desk.

Chapter 13 Submit to collection

Subject : Submit
Status : Ready
Designer : admin
Creation Date : 8/16/2007
Type : MANUAL
Description : DSpace provides an easy way for materials to enter the archive in a distributed fashion across the host institution. The submitted materials gets enter into the DSpace archive through the submission process. Materials within DSpace are always in one of three stages:

1· Items Being Assembled – the submitting user is still in the process of entering the Item's Metadata, or uploading the file(s) to be included with the item.
2· Submissions Pending Archive – the submitting user has submitted the item to the initial Collection's submission approval process. But this approval process is not yet complete, and the item is not yet archived.
3· Archived Items – the item is archived, having been approved for entry into the collection to which it was submitted.

Authorized users can initiate submissions to a DSpace collection that they have chosen. Users can choose the collection to submit to in one of two ways:

1· Explicitly, from a list of all collections in DSpace (accessible from DSpace home)
2· Contextually, by first navigating to the collection of interest (accessible from each collection's home page)

Chapter 14 Submit baseline Metadata

Subject : Submit
Status : Ready
Designer : admin
Creation Date : 8/16/2007
Type : MANUAL
Description : With their submission, users can specify baseline metadata. Users fill in the same form for all submitted items. This baseline metadata will enable users to search across all of DSpace, and to easily find submitted items in the future. The baseline metadata requested for each submitted item is based upon the qualified Dublin Core Metadata Schema, adapted to DSpace requirements.

The elements captured through the end-user interface are:
1· Author(s);
Zero or more supported. Currently no authority control for authors (i.e. DSpace does not currently know that "Samuel Clemens" and "Mark Twain" are the same author, nor does it distinguish well between two authors that share the same name).

2· Title(s);
Including alternative titles, if applicable.
3· Date of Issue
4· Series Name and Report Number; For example “Sloan School of Management Working Papers, Number 2002-128” If applicable. Zero or more supported.
5· Identifiers;
Including ISBN, ISSN, ISMN, URI, and other. Zero or more supported.
6· Language (in which submitted material is written).
7· Subject Keywords;
Zero or more supported. Currently no thesauri or authority control for subject keywords.
8· Abstract
9· Sponsors / Funding Codes
10· Other Description
Additional elements are modeled and stored internally. Some of these are generated automatically by the system. Others may be managed by DSpace administrators using the admin user interface.

Chapter 15 Submit domain specific Metadata

Subject : Submit
Status : Ready
Designer : admin
Creation Date : 8/16/2007
Type : MANUAL
Description : In addition to DSpace baseline metadata, Users can submit domain-specific metadata that is specific to their item, or that is required by the curators of the collection to which they are submitting, by uploading a file that contains the relevant metadata.

For example, a collection of images might require that each image be submitted with GIS metadata indicating the location of the corresponding image.

Users may annotate their item using as many metadata formats as desired.

DSpace calculates and maintains an MD5 checksum of each file uploaded with the item, that can be used by DSpace administrators and by users to verify the integrity of the content and metadata within the system.

DSpace will maintain each of these metadata bitstreams with the item, and make them available with the item. DSpace does not yet support discovery based upon domain-specific metadata submitted in this fashion.

Chapter 16 Upload File(s) to Item

Subject : Submit
Status : Ready
Designer : admin
Creation Date : 8/16/2007
Type : MANUAL

Description : Submitted DSpace Items are destined to become “archival atoms”, that is some amount or boundary of material that make sense together. As such, each item can include multiple pieces of content. Each piece of content might comprise several files.

Users must submit one or more files to be included with their item. For example, a user might submit a conference paper, along with presentation materials actually used at the conference. A researcher might submit a pre-print of an article, along with dataset(s) that would enable other researchers to independently reproduce the research results.

DSpace calculates and retains a checksum of each file uploaded with the item that can be used by DSpace administrators and by users to verify the integrity of the content and metadata within the system. This checksum can be obtained through the end-user interface.

DSpace attempts to recognize the bitstream format of each uploaded file. If it cannot match the bitstream format to a known format from the system’s bitstream format registry, it asks the user to describe the format so that library administrators can track and support important emerging formats over time.

Chapter 17 Grant distribution License

Subject : Submit
Status : Ready
Designer : admin
Creation Date : 8/16/2007
Type : MANUAL
Description : To enable the host institution to administer, preserve, and distribute the submitted material, DSpace asks the user to grant to the institution a non-exclusive license to distribute the material, and to translate it for the purposes of preservation.

License agreements can vary by Collection, and are specified by the collection administrator(s) for the Collection.

Because license terms are likely to change over time based on the needs of submitters and the host institution, DSpace stores a copy of the license that was granted at the time of submittal as a bitstream within the item, so that the specific terms agreed upon are always available.

Chapter 18 Augment & Approve Submissions

Subject : Submit
Status : Ready
Designer : admin
Creation Date : 8/16/2007
Type : MANUAL
Description : When the user submits an item to a DSpace collection, the system routes the submission through the approval process previously configured for that collection. The approval process can vary by collection, and can include any combination of the approval roles defined below. Further, groups of users can be used to associate any number of users with each of the roles that are configured for the process.

The system routes the submission to individuals (if any) who have been chosen to assume each of the following roles for the target collection:

1· Reviewers: review the content of the submission for appropriateness to the collection. Reviewers act as the gatekeepers for the collection. Reviewers are empowered to return a submission back to the submitter because it is deemed inappropriate for the collection. Reviewers do not edit submission metadata.
2· Approvers: check the submission for completeness and/or obvious errors (e.g. wrong file uploaded). Approvers can edit the submission's metadata to fix obvious errors, and are empowered to return a submission back to the submitter because it is incomplete or in error.
3· Metadata Editors: check and/or augment the submission's metadata. For example, a metadata editor may be assigned to add the appropriate series name and number to each submission in a collection. Metadata Editors can only edit the submission's metadata.

Each collection may specify zero or more individuals to assume each of these roles. If zero individuals are chosen, the system skips that stage of the approval process. For example, a collection might have a simple process for a series of working papers where all submissions are routed to a single staff metadata editor. This metadata editor would assign the series number to each submission. All submitted items would be approved and enter the archive – in this configuration it would be the responsibility of each submitter to crosscheck their own submissions and decide that they were suitable for the archive. Another collection might additionally define a reviewer role to ensure that submitted material is of appropriate nature to be included in the collection. Yet another collection might use all three roles. The intent is to allow communities flexibility in meeting their collection management needs while avoiding the institutional paralysis that often results from too much flexibility or too many choices.

DSpace users in each role are notified via email when they have pending submissions that require their attention. These tasks are also available to them from the "My DSpace" section of the DSpace system

As each task is completed, the submission is routed to the individual(s) responsible for the next stage of review, if any. The system sends email(s) to each of these Users, with links to their pending task in the DSpace system.

When the submission either enters the DSpace archive or is returned to the submitter, the system notifies the submitter by email about the status of their submission.

Chapter 19 My DSpace

Subject : Submit
Status : Ready
Designer : admin
Creation Date : 8/16/2007
Type : MANUAL
Description : DSpace offers each user personalized access to information within the system through their My DSpace page. As appropriate given their role(s) in the system, users can view their:

1· Items being assembled
2· submissions pending archive
3· archived Items that they submitted

4· review tasks for Items pending archive

DSpace filters these sections so that only the sections relevant to each user are presented.

1. View Archived Items Submitted by User:

Users can easily view all of the archived items that they submitted from their personalized My DSpace page.

2. View Items being Assembled

Users can access submissions that were interrupted while partially complete (because key information was not available, because all files were not available for upload, or simply because the user was called away for an extended coffee break) from the “Items being Assembled ” section of their personalized My DSpace page.

3. View Pending Approval Tasks

Users who are taking part in the approval process for one or more collections within DSpace can view, select, and perform tasks that require their attention from the “Pending Approval Tasks” section of their personalized My DSpace page.

Chapter 20 Assign/Resolve Citeable Persistent Name

Subject : Archive
Status : Ready
Designer : admin
Creation Date : 8/16/2007
Type : MANUAL
Description : DSpace creates a persistent name for submitted material that enters the archive. The name that DSpace assigns is intended to be valid and resolvable in perpetuity; the institution hosting the DSpace system stands behind that commitment, even if the DSpace system in place at the time of submission is removed, revised, or replaced. The original host institution may at some point assign that commitment to another institution (conveying responsibility for maintenance/administration of the submitted material to the other institution).

DSpace currently creates and assigns persistent names for:
· archived items

The names assigned are free of semantics about the material that they refer to. Further, they are decoupled from both logical and physical storage currently in use for associated material. Assigned names are registered using the CNRI Handle System. CNRI provides an http resolver at:

http://hdl.handle.net/<handlename>

CNRI also makes plug-ins available for most popular browsers, that allow Handles to be resolved without need for an http resolver. DSpace does not require that DSpace users install the handle plug-in, and is agnostic about its use.

The name assigned by DSpace is appropriate for citing the submitted material from other digital materials, or from print. For example, a handle assigned by the DSpace system, embedded with an http-based resolution request from CNRI, is
:
http://hdl.handle.net/1721.2/27

This work might be cited digitally or in print as:

Bass, Mick et. Al.. “DSpace – A Sustainable Solution for Digital Asset Services”. MIT DSpace, http://hdl.handle.net/1721.2/27.

Chapter 21 Store Bitstreams

Subject : Archive
Status : Ready
Designer : admin
Creation Date : 8/16/2007
Type : MANUAL
Description : DSpace provides storage for the individual bitstream(s) associated with submitted, archived, items. Users can access the stored bitstreams for any item through that item’s overview.

DSpace allows additional physical storage for bitstreams to be added, and for stored bitstreams to be easily moved from one physical store to another.

Chapter 22 Generate Bitstreams Checksums

Subject : Archive
Status : Ready
Designer : admin
Creation Date : 8/16/2007
Type : MANUAL
Description : For each bitstream maintained within the system, DSpace generates and stores an MD5 checksum that can be used either by users or by downstream preservation services to verify the integrity of the stored bitstreams over time. Checksums for each of an Item’s bitstreams are accessible from the Item Overview.

Chapter 23 Identify Bitstream Format and Preservation Service Level

Subject : Archive
Status : Ready
Designer : admin
Creation Date : 8/16/2007
Type : MANUAL
Description : DSpace maintains a bitstream format for each bitstream stored within the system, in addition to maintaining the sequence of bits associated with that bitstream. Bitstream formats

are chosen from a registry of known formats, the Bitstream Format Registry, which is maintained by DSpace administrators.

DSpace attempts to identify the bitstream format automatically. If it is unable to do so, it requests that the submitting user identify the bitstream format from those within the bitstream format registry.

If the bitstream format of a submitted bitstream has not yet been entered into the bitstream format registry, the system requests a prose description of the format, so that DSpace administrators can consider whether or not this format should be incorporated into the registry.

DSpace administrators can identify a preservation service level for each entry in the bitstream format registry.

Chapter 24 Store Item Provenance Information

Subject : Archive
Status : Ready
Designer : admin
Creation Date : 8/16/2007
Type : MANUAL
Description : DSpace maintains provenance information for each item in the archive, in qualified Dublin Core <description.provenance> values.

In addition to the system-generated provenance values described below, administrators can annotate items with additional humangenerated provenance values.

Chapter 25 Search

Subject : Disseminate
Status : Ready
Designer : admin
Creation Date : 8/16/2007
Type : MANUAL
Description : DSpace offers users the capability to search DSpace for items of interest both simple and advanced.

Chapter 26 Search Results

Subject : Disseminate
Status : Ready
Designer : admin
Creation Date : 8/16/2007
Type : MANUAL

Description : Once the user specifies a search, DSpace performs the search and produces a result set. DSpace displays the result set, including a terse description for each item in the results.
From the terse item description displayed in the search results, the user may select a desired item to view its Item Overview.

Chapter 27 Browse Items

Subject : Disseminate
Status : Ready
Designer : admin
Creation Date : 8/16/2007
Type : MANUAL
Description : From the DSpace home page, users can browse all items in DSpace by title, author, or issue date.

Chapter 28 Browse Communities and Collection

Subject : Disseminate
Status : Ready
Designer : admin
Creation Date : 8/16/2007
Type : MANUAL
Description : DSpace administrators organize DSpace items into collections, and include collections in communities. Users can browse the structure of communities and collections, in an outline view. Each outline entry includes text describing the contents and/or purpose of the corresponding community or collection. Users can select an entry from the outline view to access the home page of the selected community or collection, from which further bounded browse or search can be performed.

Chapter 29 Recent Submissions

Subject : Disseminate
Status : Ready
Designer : admin
Creation Date : 8/16/2007
Type : MANUAL
Description : Recently submitted items to each collection are displayed in a sidebar on each collection's home page.

Chapter 30 View Item Overview

Subject : Disseminate
Status : Ready
Designer : admin
Creation Date : 8/16/2007

Type : MANUAL
Description : Terse item descriptions in either the search view or browse view are linked to the corresponding item overview. Authorized users can view an item's overview, which displays the item's DSpace core metadata, lists the collection(s) that include it, and provides links to each of the bitstreams that the item comprises.

Chapter 31 Download Item Bitstream(s)

Subject : Disseminate
Status : Ready
Designer : admin
Creation Date : 8/16/2007
Type : MANUAL
Description : Users may download each of the bitstream(s) in the item by selecting the bitstream from those listed in the item's item overview.

Chapter 32 Administer Communities and Collections

Subject : Manage and Administer
Status : Ready
Designer : admin
Creation Date : 8/16/2007
Type : MANUAL
Description : 1. Create/Delete Community/Collection

Authorized DSpace users can create a new community, or create a new collection and associate it with an existing community. Admin users can also delete an existing community or collection.

2. Edit Community/Collection Home Page

Authorized users can edit key information about a community or collection. The information is used primarily within the home page of the community or collection, and includes:

· Name
· Short Description
(used in community/collection browse)
· Intro Text
· Logo (upload bitstream from local filesystem)
· Copyright Text
· Sidebar Text
Intro Text, Copyright Text, and Sidebar Text may include arbitary HTML. This allows some degree of user formatting control, and allows links to pages outside DSpace to be easily embedded.

Chapter 33 Administer Items

Subject : Manage and Administer
Status : Ready
Designer : admin
Creation Date : 8/16/2007
Type : MANUAL
Description : Authorized users can change the content and metadata of an item, or expunge the item.

1. Add/Change/Remove MetadataValues

Authorized users can use the admin user interface to:

· Add baseline metadata: that is, add to an item any number of values for any qualifer in the system's registry of Dublin Core Qualifiers.
· Change baseline metadata: that is, edit any existing value(s) for any qualifier
· Remove baseline metadata: that is, remove any/all existing value(s) for any qualifier

2. Add/Delete Bitstreams

Authorized users can use the admin user interface to:

· add to an item any number of additional bitstreams
· remove any existing bitstream(s) from the item

3. Expunge Item

· Authorized users can use the admin UI to expunge an item from the system. This deletes the item completely

Chapter 34 Administer Users and Groups

Subject : Manage and Administer
Status : Ready
Designer : admin
Creation Date : 8/16/2007
Type : MANUAL
Description : 1. Users
Authorized users can use the admin user interface to:
· Identify users who may register with DSpace. The email address of valid users must be entered in this way before users will be able to register for the system.

· Edit the personal information stored for a specified user, including their email, first name, last name, phone, password, and whether or not they are active in DSpace.

· Delete a specified user from the system.

2. Groups

Authorized users can create and maintain groups of users. DSpace uses these groups and authorization policies to determine which users are authorized to perform particular actions within the system.

Authorized users can use the admin user interface to:
· Create a group of users
· Name the group
· Add users to the group
· Remove users from the group

Chapter 35 Authenticate / Authorize users

Subject : Manage and Administer
Status : Ready
Designer : admin
Creation Date : 8/16/2007
Type : MANUAL
Description : DSpace uses a set of machine-interpretable policy statements to determine whether or not a given user is authorized to perform a given action upon a given resource.

Chapter 36 List / Abort Submission Processes

Subject : Manage and Administer
Status : Ready
Designer : admin
Creation Date : 8/16/2007
Type : MANUAL
Description : Authorized users can view the state of open submission processes in the system, and can abort a selected submission process.

When a submission process is aborted, any pending tasks are removed from the “My DSpace” tasklists of approvers, reviewers, and/or metadata editors. The submitted item remains accessible to the submitter from their “My DSpace” page.

Chapter 37 Manage Bitsream Format Registry

Subject : Manage and Administer
Status : Ready
Designer : admin
Creation Date : 8/16/2007
Type : MANUAL
Description : Authorized users can add, edit, or delete a format in the system’s registry of bitstream formats. DSpace uses this registry to allow users to identify the format of bitstreams they submit to DSpace.

Chapter 38 Bulk Import

Subject : Manage and Administer
Status : Ready
Designer : admin
Creation Date : 8/16/2007
Type : MANUAL
Description : DSpace system administrators can bulk import items into the system, and bulk identify users to the system. Unlike all of the preceding administrative functionality, the bulk import tools are not available in the web-client-based administrative user interface. Rather, bulk import tools are driven from a command-line interface.

Chapter 39 Bulk Identify Users

Subject : Manage and Administer
Status : Ready
Designer : admin
Creation Date : 8/16/2007
Type : MANUAL
Description : DSpace administrators can bulk identify users to the system through LDAP authentication configuration.

Chapter 40 Run Tests

Chapter 41 Test Set : DSpace_functional_te

Status : Closed
Open Date : 7/2/2007
Close Date : 8/27/2007

Chapter 42 Test: Site

Test Set : DSpace_functional_te
Status : Passed
Tester : admin
Exec Date : 8/27/2007
Time : 23:40:45

Chapter 43 Run Name : Run(8/27 23:39:53)

Status : Passed
Tester Name : admin
Exec Date : 8/27/2007
Exec Time : 23:39:53

Run Steps:

Name	Status	Description	Expected	Actual
Step 1	Passed	Open the browser and put the url http://localhost/dspace	Our installed DSpace Site should open.	DSpace Site openned successfully

Chapter 44 Run Name : Run(8/27 23:40:45)

Status : Passed
Tester Name : admin
Exec Date : 8/27/2007
Exec Time : 23:40:45

Run Steps:

Name	Status	Description	Expected	Actual
Step 1	Passed	Open the browser and put the url http://localhost/dspace	Our installed DSpace Site should open.	DSpace Site opened successfully

Chapter 45 Test: User

Test Set : DSpace_functional_te
Status : Passed
Tester : admin
Exec Date : 8/27/2007
Time : 23:41:21

Chapter 46 Run Name : Run(8/27 23:41:21)

Status : Passed
Tester Name : admin
Exec Date : 8/27/2007
Exec Time : 23:41:21

Run Steps:

Name	Status	Description	Expected	Actual
Step 1	Passed	Open the browser and write the url of the DSpace site. Some thing like: http://yourhost/dspace	DSpace site should be accessible	site is accessible
Step 2	Passed	Anonymous user start searching browse and download the file(s) from the items of the collection in the DSpace	Searching, Browsing and Downoad is possible.	Searching, Browsing and Downoad is possible successfully.
Step 3	Passed	Registered user logon to the system by clicking on My DSpace	Should logon successfully	Registered user can logon to the system successfully
Step 4	Passed	Registered user can start submission in the collection of DSpace	Submission process should be completed successfully.	Submission process completed successfully.

Chapter 47 Test: Group of Users

Test Set : DSpace_functional_te
Status : Passed
Tester : admin
Exec Date : 8/27/2007
Time : 23:44:19

Chapter 48 Run Name : Run(8/27 23:44:19)

Status : Passed
Tester Name : admin
Exec Date : 8/27/2007
Exec Time : 23:44:19

Run Steps:

Name	Status	Description	Expected	Actual
Step 1	Passed	DSpace administrator should logged in and and open the admin page with the URL http://yourhost/dspace/dspace-admin	Admin page of DSpace should open	Admin page of DSpace opened successfully
Step 2	Passed	Add E-People into the system	E-People should be added successfully	E-People added successfully
Step 3	Passed	Add Groups by selecting the E-People	Group should be formed successfully	Group formed successfully

Chapter 49 Test: Community

Test Set : DSpace_functional_te
Status : Passed
Tester : admin
Exec Date : 8/27/2007
Time : 23:45:30

Chapter 50 Run Name : Run(8/27 23:45:30)

Status : Passed
Tester Name : admin
Exec Date : 8/27/2007
Exec Time : 23:45:30

Run Steps:

Name	Status	Description	Expected	Actual
Step 1	Passed	Logged on to the system and click on "Communities & Collections".	"Communities & Collections" home page should open.	"Communities & Collections" home page opened.
Step 2	Passed	To create a new "Community" click on "Create Top-Level Community" button on the right hand side below "Admin Tools"	Create Community blank form should appear	Appeared
Step 3	Passed	Fill up the Create Community form by providing the community name other relevant information and click on "Create" button.	Community should be created	Created successfully

Chapter 51 Test: Collection

Test Set : DSpace_functional_te
Status : Passed
Tester : admin
Exec Date : 8/27/2007
Time : 23:46:44

Chapter 52 Run Name : Run(8/27 23:46:44)

Status : Passed
Tester Name : admin
Exec Date : 8/27/2007
Exec Time : 23:46:44

Run Steps:

Name	Status	Description	Expected	Actual
Step 1	Passed	To create collection in DSpace select the community under which the collection has to formed.	Community home page should open	Community home page opened successfully
Step 2	Passed	In the community home page click on "Create Collection" in the right hand corner under the Admin Tools	Describe collection windows should appear.	Appeared
Step 3	Passed	After describing the collection select E-People and Groups for authorization to submit into the collection.	E-People and Groups should be authorized and "Edit collection page" should open.	Edit collectio page appeared
Step 4	Passed	Fill up the Edit collection form and click on "update"	New collection should be created updated.	Created successfully

Chapter 53 Test: Approval Process

Test Set : DSpace_functional_te
Status : Passed
Tester : admin
Exec Date : 8/27/2007
Time : 23:54:58

Chapter 54 Run Name : Run(8/27 23:54:58)

Status : Passed
Tester Name : admin
Exec Date : 8/27/2007
Exec Time : 23:54:58

Run Steps:

Name	Status	Description	Expected	Actual
Step 1	Passed	Submit items in the collection which has some approval workflow	Mail should be generated to the approver of the collection	Mail generated to the approver of the collection successfully
Step 2	Passed	Approver should logon to the system	Approver could see the pending task to approve the submission	Approver can see the pending task to approve the submission

Step 3	Passed	Approvr click on the submitted item to view in order to accept/reject the same	submitted item either accepted/rejected by the approver	submitted item can be successfully either accepted/rejected by the approver

Chapter 55 Test: Item

Test Set : DSpace_functional_te
Status : Passed
Tester : admin
Exec Date : 8/27/2007
Time : 23:56:52

Chapter 56 Run Name : Run(8/27 23:56:52)

Status : Passed
Tester Name : admin
Exec Date : 8/27/2007
Exec Time : 23:56:52

Run Steps:

Name	Status	Description	Expected	Actual
Step 1	Passed	Select the Communities and the collection in which item has to be submitted	Collection home page should open	Collection home page opened successfully
Step 2	Passed	Click on "Submit to this Collection" button on the collection home page	Submit: "Describe your Item" page should appear	Appeared
Step 3	Passed	Upload File(s) to the Item and click on save	verify bitstream and checksum page should appear	Appeared successfully
Step 4	Passed	Verify the item and click on next to accept the license	Item submission should be completed	Item submission completed successfully

Chapter 57 Test: Anonymous vs Credential Access

Test Set : DSpace_functional_te
Status : Passed
Tester : admin
Exec Date : 8/28/2007
Time : 00:04:51

Chapter 58 Run Name : Run(8/28 0:4:51)

Status : Passed
Tester Name : admin
Exec Date : 8/28/2007
Exec Time : 00:04:51

Run Steps:

Name	Status	Description	Expected	Actual
Step 1	Passed	As anonymous user make search, browse and download the file(s) in the items.	Any anonymous user should be able to search browse and download file(s) in DSpace	Anonymous user can search browse and download file(s) in DSpace
Step 2	Passed	As anonymous user click on My DSpace	System should prompt to login	Prompted
Step 3	Passed	Login into the system and submit item to the collection	Item should be submitted successfully	Item submitted successfully

Chapter 59 Test: Register with DSpace

Test Set : DSpace_functional_te
Status : Passed
Tester : admin
Exec Date : 8/28/2007
Time : 00:07:12

Chapter 60 Run Name : Run(8/28 0:7:12)

Status : Passed
Tester Name : admin
Exec Date : 8/28/2007
Exec Time : 00:07:12

Run Steps:

Name	Status	Description	Expected	Actual
Step 1	Passed	Click on My DSpace	Log In to DSpace window should appear	Appeared
Step 2	Passed	Click on "New user? Click here to Register".	User Registration page should appear	Appeared
Step 3	Passed	Write email address and click on "Register" button	Mail should be generated and link should be sent register and login to the system	Mail generated with the link successfully

Chapter 61 Test: Forgotten Password

Test Set : DSpace_functional_te
Status : Passed
Tester : admin
Exec Date : 8/28/2007
Time : 00:08:10

Chapter 62 Run Name : Run(8/28 0:8:10)

Status : Passed
Tester Name : admin
Exec Date : 8/28/2007
Exec Time : 00:08:10

Run Steps:

Name	Status	Description	Expected	Actual
Step 1	Passed	Click on "My DSpace"	Log In to DSpace window should open	Lonin window opened
Step 2	Passed	Click on "Have you forgotten your password" link	Forgotten Password window should appear	Appeared
Step 3	Passed	Write the email id of registered user and click	Email will be generated and link will be sent to	Email and link sent successfully

		on "I Forget My Password"	reset the password	

Chapter 63 Test: Edit user Profile

Test Set : DSpace_functional_te
Status : Passed
Tester : admin
Exec Date : 8/28/2007
Time : 00:09:39

Chapter 64 Run Name : Run(8/28 0:9:39)

Status : Passed
Tester Name : admin
Exec Date : 8/28/2007
Exec Time : 00:09:39

Run Steps:

Name	Status	Description	Expected	Actual
Step 1	Passed	Registered user logon to the system	Should be able to Login successfully	Successfully logged in
Step 2	Passed	Click on "Edit Profile"	Should get "Edit Your Profile" page.	"Edit Your Profile" page appeared.
Step 3	Passed	Fill up the form with "First Name", Last Name", "Contact Telephobe", set newpassword if needed and click on Update Profile.	User's Profile should be updated	User's Profile updated successfully

Chapter 65 Test: Submit to collection

Test Set : DSpace_functional_te

Status : Passed
Tester : admin
Exec Date : 8/28/2007
Time : 00:11:26

Chapter 66 Run Name : Run(8/28 0:11:26)

Status : Passed
Tester Name : admin
Exec Date : 8/28/2007
Exec Time : 00:11:26

Run Steps:

Name	Status	Description	Expected	Actual
Step 1	Passed	Registered should logon to the system	Should be able to login.	Loged in successfully
Step 2	Passed	Select the Comunity and the collection in which item has to be submitted	Should get the colection home page window	Collection home page window appeared
Step 3	Passed	Click on "Submit to This Collection" button on the collection home page.	Submission process should start	submission process started
Step 4	Passed	Describe the item and upload the file(s) in the item which is being submitted	file(s) should be uploaded	files uploaded successfully
Step 5	Passed	Verify the item submitted and grant the license	Submission should complete successfully	Submission completed successfully

Chapter 67 Test: Submit baseline Metadata

Test Set : DSpace_functional_te
Status : Passed
Tester : admin
Exec Date : 8/28/2007
Time : 00:13:16

Chapter 68 Run Name : Run(8/28 0:13:16)

Status : Passed
Tester Name : admin

Exec Date : 8/28/2007
Exec Time : 00:13:16

Run Steps:

Name	Status	Description	Expected	Actual
Step 1	Passed	Logon to the DSpace and start submittin "Item" after selecting "Community and Collection"	Submission Process should start	Submission process started
Step 2	Passed	Describe the Item baseline metadata such as: Author(s); · Title(s); · Date of Issue · Series Name and Report Number · Identifiers; · Language (in which submitted material is written). · Subject Keywords; · Abstract · Sponsors / Funding Codes · Other Description and click on "Next" button.	User should be able to define Baseline Metadata for the item submission and get the Document file upload page.	Baseline Metadata defined successfully and document has been uploaded

Chapter 69 Test: Submit domain specific Metadata

Test Set : DSpace_functional_te
Status : Passed
Tester : admin
Exec Date : 8/28/2007
Time : 00:15:17

Chapter 70 Run Name : Run(8/28 0:15:17)

Status : Passed
Tester Name : admin
Exec Date : 8/28/2007
Exec Time : 00:15:17

Run Steps:

Name	Status	Description	Expected	Actual
Step 1	Passed	Select the submitted item and click on "Edit"	Edit Item page should open	Edit Item page opened successfully
Step 2	Passed	In the Edit Item page submit domain specific metadata and click on "uipdate" button at the bottom.	Domain specific Metadata should be submitted and updated	Domain specific Metadata has been submitted and updated successfully

Chapter 71 Test: Upload File(s) to Item

Test Set : DSpace_functional_te
Status : Passed
Tester : admin
Exec Date : 8/28/2007
Time : 00:16:29

Chapter 72 Run Name : Run(8/28 0:16:29)

Status : Passed
Tester Name : admin
Exec Date : 8/28/2007
Exec Time : 00:16:29

Run Steps:

Name	Status	Description	Expected	Actual
Step 1	Passed	Logon to the system and start submitting an item	Submission process should start	Submission process started
Step 2	Passed	Describe the item and click on "Next" to upload the "Document File"	Document File should be uploaded	Document File uploaded successfully

Chapter 73 Test: Grant distribution License

Test Set : DSpace_functional_te
Status : Passed
Tester : admin
Exec Date : 8/28/2007
Time : 00:17:33

Chapter 74 Run Name : Run(8/28 0:17:33)

Status : Passed
Tester Name : admin
Exec Date : 8/28/2007
Exec Time : 00:17:33

Run Steps:

Name	Status	Description	Expected	Actual
Step 1	Passed	While submitting an item after describing, uploading file(s) verification click on next	License page should display	License page displayed
Step 2	Passed	In the bottom of License page click on the button "I Grant the License"	License should be granted and submission should be completed	License has been granted and submission has been completed

Chapter 75 Test: Augment & Approve Submissions

Test Set : DSpace_functional_te
Status : Passed
Tester : admin
Exec Date : 8/28/2007
Time : 00:18:52

Chapter 76 Run Name : Run(8/28 0:18:52)

Status : Passed
Tester Name : admin
Exec Date : 8/28/2007
Exec Time : 00:18:52

Run Steps:

Name	Status	Description	Expected	Actual
Step 1	Passed	After user submitted an item a mail is generated to reviewer(s) and he/she needs to logon to the system	After login the reviewer can see his pending task	Reviewer is able to view his pending task
Step 2	Passed	Now reviewer can review the submitted item and can accept it or can return it back to the submitter	Reviewer task is completed	Reviewer task is successfully completed
Step 3	Passed	After the review has been done mail will be sent to the approvers to approve/reject the submission. For the same he/she needs to logon to the system and can approve/reject the submission.	Approver task is completed	Approver task is successfully completed
Step 4	Passed	After the item has been approved mail will be sent to the metadata editors. He/She can logon to the system and can edit the Metadata	Metadat editors task is completed	Metadat editors task is successfully completed

Chapter 77 Test: My DSpace

Test Set : DSpace_functional_te
Status : Passed
Tester : admin
Exec Date : 8/28/2007
Time : 00:20:51

Chapter 78 Run Name : Run(8/28 0:20:51)

Status : Passed
Tester Name : admin
Exec Date : 8/28/2007
Exec Time : 00:20:51

Run Steps:

Name	Status	Description	Expected	Actual
Step 1	Passed	Click on "My DSpace"	System will ask to login	Login page appeared
Step 2	Passed	Logon to the system	user should see their respective pending tasks depending on their roles	pending task appeared

Chapter 79 Test: Assign/Resolve Citeable Persistent Name

Test Set : DSpace_functional_te
Status : Passed
Tester : admin
Exec Date : 8/28/2007
Time : 00:21:48

Chapter 80 Run Name : Run(8/28 0:21:48)

Status : Passed
Tester Name : admin
Exec Date : 8/28/2007
Exec Time : 00:21:48

Run Steps:

Name	Status	Description	Expected	Actual

Step 1	Passed	Mention the handle.prefix acquired by the CNRI in the dspace.cfg file:
		
		Some thing like this
		
		##### Handle settings ######
		
		# CNRI Handle prefix
		handle.prefix = 123456789
			The same prefix should appear in URI prefix of the item submited to dspace.	Same prefix appeared

Chapter 81 Test: Store Bitstreams

Test Set : DSpace_functional_te
Status : Passed
Tester : admin
Exec Date : 8/28/2007
Time : 00:22:24

Chapter 82 Run Name : Run(8/28 0:22:24)

Status : Passed
Tester Name : admin
Exec Date : 8/28/2007
Exec Time : 00:22:24

Run Steps:

Name	Status	Description	Expected	Actual
Step 1	Passed	To add additional bitstream for any item. In the item overview page click on edit button.	"Edit Item" page should open.	"Edit Item" page opened.
Step 2	Passed	Click on "Add Bitstream" button and browse and upload the Bitstream	New Bitstream should be added.	New Bitstream is added.

Chapter 83 Test: Generate Bitstreams Checksums

Test Set : DSpace_functional_te
Status : Passed
Tester : admin
Exec Date : 8/28/2007
Time : 00:23:57

Chapter 84 Run Name : Run(8/28 0:23:22)

Status : Not Completed
Tester Name : admin
Exec Date : 8/28/2007

Exec Time : 00:23:22

Run Steps:

Name	Status	Description	Expected	Actual
Step 1	No Run	While submittint the item after uploading the file(s) we can click on the button "Show checksums"	"Check sum" generated should match	"Check sum" generated matched

Chapter 85 Run Name : Run(8/28 0:23:57)

Status : Passed
Tester Name : admin
Exec Date : 8/28/2007
Exec Time : 00:23:57

Run Steps:

Name	Status	Description	Expected	Actual
Step 1	Passed	While submittint the item after uploading the file(s) we can click on the button "Show checksums"	"Check sum" generated should match	"Check sum" generated matched

Chapter 86 Test: Identify Bitstream Format and Preservation Service Level

Test Set : DSpace_functional_te
Status : Passed
Tester : admin
Exec Date : 8/28/2007
Time : 00:24:29

Chapter 87 Run Name : Run(8/28 0:24:29)

Status : Passed
Tester Name : admin
Exec Date : 8/28/2007
Exec Time : 00:24:29

Run Steps:

Name	Status	Description	Expected	Actual
Step 1	Passed	Submit new item and upload the file(s) into the item	DSpace attempts to identify the bitstream format automatically. If it is unable to do so, it requests that the submitting user identify the bitstream format from those within the bitstream format registry.	Bitstream identified
Step 2	Passed	If the bitstream format of a submitted bitstream has not yet been entered into the bitstream format registry.	The system requests a prose description of the format so that DSpace administrators can consider whether or not this format should be incorporated into the registry.	Requested

Chapter 88 Test: Store Item Provenance Information

Test Set : DSpace_functional_te
Status : Passed
Tester : admin
Exec Date : 8/28/2007
Time : 00:26:55

Chapter 89 Run Name : Run(8/28 0:26:55)

Status : Passed
Tester Name : admin
Exec Date : 8/28/2007
Exec Time : 00:26:55

Run Steps:

Name	Status	Description	Expected	Actual
Step 1	Passed	To add or edit stred item provenance item. Click on Edit button in the item overview page	Edit Item page should open	Edit Item page opened
Step 2	Passed	New provenance can be annotated by selecting	Provenace should be added by clicking on add	Added

		Element, Qualifier, Value and Launguage.	button.	

Chapter 90 Test: Search

Test Set : DSpace_functional_te
Status : Passed
Tester : admin
Exec Date : 8/28/2007
Time : 00:27:29

Chapter 91 Run Name : Run(8/28 0:27:29)

Status : Passed
Tester Name : admin
Exec Date : 8/28/2007
Exec Time : 00:27:29

Run Steps:

Name	Status	Description	Expected	Actual
Step 1	Passed	Write the search term and click on "Go" in the top left hand side "Search DSpace" for simple search	DSpace does the free text search and displays the results.	Result is displayed
Step 2	Passed	For advance search click on the "Advance Search" below the search text box.	Advance Search box should open	Advance search box appeared
Step 3	Passed	Select the "Search Type", write the term "Search For" and refine the search through boolean operators "And/or/Not"	Search Result should appear by clicking on "Search" button.	Search Result displayed successfully

Chapter 92 Test: Search Results

Test Set : DSpace_functional_te
Status : Passed
Tester : admin
Exec Date : 8/28/2007
Time : 00:28:40

Chapter 93 Run Name : Run(8/28 0:28:40)

Status : Passed
Tester Name : admin
Exec Date : 8/28/2007
Exec Time : 00:28:40

Run Steps:

Name	Status	Description	Expected	Actual
Step 1	Passed	Write the search term and click on "Search" button.	Search result should display.	Search result displayed.

Chapter 94 Test: Browse Items

Test Set : DSpace_functional_te
Status : Passed
Tester : admin
Exec Date : 8/28/2007
Time : 00:31:30

Chapter 95 Run Name : Run(8/28 0:31:30)

Status : Passed
Tester Name : admin
Exec Date : 8/28/2007
Exec Time : 00:31:30

Run Steps:

Name	Status	Description	Expected	Actual
Step 1	Passed	Select the community, sub-community and the collection whose Items we want to browse	Collection homepage should open	Collection homepage opened

Step 2	Passed	Write the search term and click on "Go" button	Hit items should display	Hit items displayed
Step 3	Passed	We can also browse the list of items in a collection by Titles, Authors, Subjects, By Date.	List of items present in the collection should display.	List of items displayed.

Chapter 96 Test: Browse Communities and Collection

Test Set : DSpace_functional_te
Status : Passed
Tester : admin
Exec Date : 8/28/2007
Time : 00:32:59

Chapter 97 Run Name : Run(8/28 0:32:59)

Status : Passed
Tester Name : admin
Exec Date : 8/28/2007
Exec Time : 00:32:59

Run Steps:

Name	Status	Description	Expected	Actual
Step 1	Passed	Open the Internet Browser and write the URL the DSpace Digital Library.	DSpace Digital Library home page should open	DSpace Digital Library home page opened
Step 2	Passed	Click on "Communities and Collection" in left hand side bar under the "Browse" option	Communitites and collection homepage should open where the list of communities, subcommunities and collection within should display	List displayed

Chapter 98 Test: Recent Submissions

Test Set : DSpace_functional_te
Status : Passed
Tester : admin
Exec Date : 8/28/2007
Time : 00:34:03

Chapter 99 Run Name : Run(8/28 0:34:3)

Status : Passed
Tester Name : admin
Exec Date : 8/28/2007
Exec Time : 00:34:03

Run Steps:

Name	Status	Description	Expected	Actual
Step 1	Passed	Select the collection and click to open the collection home page	Recent submission to the collection should display	Recent submission is displayed

Chapter 100 Test: View Item Overview

Test Set : DSpace_functional_te
Status : Passed
Tester : admin
Exec Date : 8/28/2007
Time : 00:34:40

Chapter 101 Run Name : Run(8/28 0:34:40)

Status : Passed
Tester Name : admin
Exec Date : 8/28/2007
Exec Time : 00:34:40

Run Steps:

Name	Status	Description	Expected	Actual
Step 1	Passed	Search or Browse for the items in the DSpace and select the item	overview of the selected item should display	Displayed

Chapter 102 Test: Download Item Bitstream(s)

Test Set : DSpace_functional_te
Status : Passed
Tester : admin
Exec Date : 8/28/2007
Time : 00:35:01

Chapter 103 Run Name : Run(8/28 0:35:1)

Status : Passed
Tester Name : admin
Exec Date : 8/28/2007
Exec Time : 00:35:01

Run Steps:

Name	Status	Description	Expected	Actual
Step 1	Passed	Select the Item from the list present in the collection and click on it	Item overview page should open	Item overview page opened successfully
Step 2	Passed	Select the bitstream and Click on "View / Open"	Downolad of the selected bitstream should start	Downloading starts

Chapter 104 Test: Administer Communities and Collections

Test Set : DSpace_functional_te
Status : Passed
Tester : admin
Exec Date : 8/28/2007
Time : 00:36:07

Chapter 105 Run Name : Run(8/28 0:36:7)

Status : Passed
Tester Name : admin
Exec Date : 8/28/2007
Exec Time : 00:36:07

Run Steps:

Name	Status	Description	Expected	Actual
Step 1	Passed	To Create a new Community/Collection login as authorized user and click on "Create Top-Level Community" or "Collection"	Create Community page should open.	Create community page opened
Step 2	Passed	Fill the details of the community/Collection, upload logo, edit authorization and click on create button.	New community/collection should be created	Created
Step 3	Passed	To delet a community/collection go to the respective community/collection home page and click on "Edit"	Edit page should open	Opened
Step 4	Passed	Click on "Delete this Community" or "Delete this collection" button.	system will ask to "Delete" or "Cancel" click on "delete" button to finally delete the community/collection.	Deleted
Step 5	Passed	To edit the community/collection go to the community/collection home page and click on edit.	Edit page open, edit the details and click on "Update".	Edit page opened

Chapter 106 Test: Administer Items

Test Set : DSpace_functional_te
Status : Passed
Tester : admin
Exec Date : 8/28/2007
Time : 00:37:41

Chapter 107 Run Name : Run(8/28 0:37:41)

Status : Passed
Tester Name : admin
Exec Date : 8/28/2007
Exec Time : 00:37:41

Run Steps:

Name	Status	Description	Expected	Actual
Step 1	Passed	Search or browse the item and open its overview, click on edit button	Edit Item page should open. In this page: Add/Change/Remove MetadataValues, Add/Delete Bitstreams and Expunge Item can be done	Edit page appeared and actions performed smoothly

Chapter 108 Test: Administer Users and Groups

Test Set : DSpace_functional_te
Status : Passed
Tester : admin
Exec Date : 8/28/2007
Time : 00:38:33

Chapter 109 Run Name : Run(8/28 0:38:33)

Status : Passed
Tester Name : admin
Exec Date : 8/28/2007
Exec Time : 00:38:33

Run Steps:

Name	Status	Description	Expected	Actual
Step 1	Passed	Login as dspace-admin and click on E-People	Administer EPeople page should open	Administer EPeople page opened
Step 2	Passed	In the Administer EPeople click on Add EPerson to add a user.	Edit EPerson page should open. Fill the details and click on save to create a new user.	Page opened and new user has been created
Step 3	Passed	To edit/delete E-Person	E-Person should be edited	Successfully

		click on "Select E-Person" select the person and then click on edit or detete button.	or deleted	
Step 4	Passed	Select Groups as an DSpace-admin	Group editor page should open	Page appeared
Step 5	Passed	To create a new group in DSpace click on "Create New Group" button	Edit new group page should open.	Edit new group page opened.
Step 6	Passed	Give the group name and select the E-Person Members and Group Members and click on "Update Groups"	New Group should be added.	New Group added successfully

Chapter 110 Test: Authenticate / Authorize users

Test Set : DSpace_functional_te
Status : Passed
Tester : admin
Exec Date : 8/28/2007
Time : 00:41:03

Chapter 111 Run Name : Run(8/28 0:41:3)

Status : Passed
Tester Name : admin
Exec Date : 8/28/2007
Exec Time : 00:41:03

Run Steps:

Name	Status	Description	Expected	Actual
Step 1	Passed	To administer the authorization policies click on "Authorization" as Dspace-admin	Administer authorization policies page should open. Here administrator can choose resource to manage policies	Page opened successfully

Chapter 112 Test: List / Abort Submission Processes

Test Set : DSpace_functional_te
Status : Passed
Tester : admin
Exec Date : 8/28/2007
Time : 00:42:17

Chapter 113 Run Name : Run(8/28 0:42:17)

Status : Passed
Tester Name : admin
Exec Date : 8/28/2007
Exec Time : 00:42:17

Run Steps:

Name	Status	Description	Expected	Actual
Step 1	Passed	Select the collection and click on edit to view its submission process	Here the authorized users can view List/abort the submission process.	Authorized users can view List/abort the submission process

Chapter 114 Test: Manage Bitsream Format Registry

Test Set : DSpace_functional_te
Status : Passed
Tester : admin
Exec Date : 8/28/2007
Time : 00:43:26

Chapter 115 Run Name : Run(8/28 0:43:26)

Status : Passed
Tester Name : admin
Exec Date : 8/28/2007
Exec Time : 00:43:26

Run Steps:

Name	Status	Description	Expected	Actual
Step 1	Passed	Select Bitstream Format Registry as DSpace-admin.	List of Bitstream Format Registry page should display. Here Authorized users can add, edit, or delete a format in the system's registry of bitstream formats.	List of Bitstream format page appeared

Chapter 116 Test: Bulk Import

Test Set : DSpace_functional_te
Status : Failed
Tester : admin
Exec Date : 8/28/2007
Time : 01:01:21

Chapter 117 Run Name : Run(8/28 0:44:15)

Status : Not Completed
Tester Name : admin
Exec Date : 8/28/2007
Exec Time : 00:44:15

Chapter 118 Run Name : Run(8/28 1:1:21)

Status : Failed
Tester Name : admin
Exec Date : 8/28/2007
Exec Time : 01:01:21

Run Steps:

Name	Status	Description	Expected	Actual
Step 1	Failed	Bulk import the users as well items from the UI	Should get imported	Not possible

Chapter 119 Test: Bulk Identify Users

Test Set : DSpace_functional_te
Status : Passed
Tester : admin
Exec Date : 8/28/2007
Time : 00:45:08

Chapter 120 Run Name : Run(8/28 0:45:8)

Status : Passed
Tester Name : admin
Exec Date : 8/28/2007
Exec Time : 00:45:08

Run Steps:

Name	Status	Description	Expected	Actual
Step 1	Passed	Open dspace.cfg file and edit the LDAP Authentication configuration settings such as: ldap.enable = true ldap.provider_url = ldap://ldap.myu.edu/o=myu.edu ldap.id_field = uid ldap.object_context = ou=people,o=myu.edu ldap.search_context = ou=people ldap.email_field = mail ldap.surname_field = sn ldap.givenname_field = givenName ldap.phone_field = telephoneNumber	Binding should be done	Binding has been done successfully
Step 2	Passed	After binding enable webui.ldap.autoregister in dspace.cfg ## webui.ldap.autoregister ## webui.ldap.autoregister = true	LDAP users should login into the system	LDAP users can login into the system successfully

Chapter 121 Track Defects

Chapter 122 Defect ID : 2

Status : Open
Assigned To : admin
Project : Project
Subject : DSpace
Summary : Currently no authority
Reproducible : Y
Severity : 3-High
Priority : 3-High
Detected By : ehtesham
Detected on Date : 8/17/2007
Description : DSpace does not currently know that “Samuel Clemens” and “Mark Twain” are the same author, nor does it distinguish well between two authors that share the same name).

History:

Field Name	Change Date	Change Time	Changer	New Value
Status	8/17/2007	07:37:38	admin	
Assigned To	8/17/2007	07:37:38	admin	
Status	8/28/2007	00:48:45	admin	Open
Assigned To	8/28/2007	00:48:45	admin	admin

Chapter 123 Defect ID : 5

Status : Open
Assigned To : admin
Project : Project
Subject : DSpace
Summary : Version Control
Reproducible : Y
Severity : 3-High
Priority : 3-High
Detected By : admin
Test Set : DSpace_functional_te
Detected on Date : 8/28/2007
Description : Feature of "Version Control" is absent.

History:

Field Name	Change Date	Change Time	Changer	New Value
Status	8/28/2007	00:54:43	admin	Open
Assigned To	8/28/2007	00:54:43	admin	admin

Chapter 124 Defect ID : 3

Status : Open
Assigned To : admin
Project : Project
Subject : DSpace
Summary : Currently no thesauri or authority control for subject keywords.
Reproducible : Y
Severity : 3-High
Priority : 3-High
Detected By : ehtesham
Detected on Date : 8/17/2007
Description : Currently no thesauri or authority control for subject keywords.

History:

Field Name	Change Date	Change Time	Changer	New Value
Status	8/17/2007	07:38:47	admin	
Assigned To	8/17/2007	07:38:47	admin	
Status	8/28/2007	00:49:14	admin	Open
Assigned To	8/28/2007	00:49:14	admin	admin

Chapter 125 Defect ID : 4

Status : Open
Assigned To : admin
Project : Project
Subject : DSpace
Summary : Priotitized List display of Communitites and collection
Reproducible : Y
Severity : 3-High
Priority : 3-High
Detected By : ehtesham
Detected on Date : 8/17/2007
Description : Communities and collection does display in prioritized listing. It gets sorted alphabetically.

History:

Field Name	Change Date	Change Time	Changer	New Value
Status	8/17/2007	08:01:00	admin	
Assigned To	8/17/2007	08:01:00	admin	
Status	8/28/2007	00:52:59	admin	Open
Assigned To	8/28/2007	00:52:59	admin	admin

Chapter 126 Defect ID : 6

Status : Open
Assigned To : admin
Project : Project

Subject : DSpace
Summary : User interface bulk import
Reproducible : Y
Severity : 3-High
Priority : 3-High
Detected By : admin
Test Set : DSpace_functional_te
Detected on Date : 8/28/2007
Description : User interface bulk import options (users as well as items) are not present

History:

Field Name	Change Date	Change Time	Changer	New Value
Status	8/28/2007	00:57:35	admin	Open
Assigned To	8/28/2007	00:57:35	admin	admin

Index

S

U

V