Lab session 02: Understanding the concepts of Software Documentation.

Basic Worth of Documentation

- All large software development projects, irrespective of application, generate a large amount of associated documentation.
- Infact For moderately sized systems, the documentation will probably fill several filing cabinets; for large systems, it may fill several rooms. A high proportion of software process costs is incurred in producing this documentation.
- Documentation errors and lapses can lead to errors by end-users and consequent system failures with their associated costs and disruption. Therefore, managers and software engineers should pay as much attention to documentation and its associated costs as to the development of the software itself.

How Document should be

- They should act as a communication medium between members of the development team & later on Maintenance team.
- They should provide information for management to help them plan, budget and schedule the software development process.
- Some of the documents should tell users how to use and administer the system in short training documents .
  - So *from informal working documents* to *professionally produced user manuals* we are facing standard documentation.
  - Software engineers are usually responsible for producing most of this documentation although professional technical writers may assist with the final polishing of externally released information.

SRS

- A Software Requirements Specification (SRS) is a document that describes the nature of a project, software or application.
- In simple words, SRS document is a manual of a project before you start a project/application.
- This document is also known by the names SRS report, software document.
- There are a set of guidelines to be followed while preparing the software requirement specification document.
- This Basically includes the purpose, scope, functional, nonfunctional, software and hardware requirements of the project.
- In addition to this, it also contains the information about environmental conditions required, safety and security requirements, software quality attributes of the project etc.
- It also includes the yield and cost of the software.
- In this Lab, flight management project is used as an example to explain how ***generally*** SRS could look like

SRS

General Table of Contents for a SRS Document:

1. Introduction
   - 1.1 Purpose
   - 1.2 Document Conventions
   - 1.3 Intended Audience and Reading Suggestions
   - 1.4 Project Scope
   - 1.5 References
2. Overall Description
   - 2.1 Product Perspective
   - 2.2 Product Features
   - 2.3 User Classes and Characteristics
   - 2.4 Operating Environment
   - 2.5 Design and Implementation Constraints
   - 2.6 Assumptions and Dependencies

General Table of Contents for a SRS Document (cont.)

1. Introduction

1.1 PURPOSE

The purpose of this document is to build an online system to manage flights and passengers to ease the flight management. <<*Include the purpose as applicable to your project* >>

1.2 DOCUMENT CONVENTIONS

This document uses the following conventions.(use conventions as per your application)

Convention	Meaning
DB	Database
DDB	Distributed Database
ER	Entity Relationship

1.3 INTENDED AUDIENCE AND READING SUGGESTIONS

This project is a prototype for the flight management system and it is restricted with in the university premises. This has been implemented under the guidance of University faculty. This project is useful for the flight management team and as well as to the passengers.

1. Introduction(cont.)

1.4 PROJECT SCOPE

The purpose of the online flight management system is to ease flight management and to create a convenient and easy-to-use application for passengers, trying to buy airline tickets.
The system is based on a relational database with its flight management and reservation functions. We will have a database server supporting hundreds of major cities around the world as well as thousands of flights by various airline companies. Above all, we hope to provide a comfortable user experience along with the best pricing available.

1.5 REFERENCES

- https://krazytech.com/projects
- Fundamentals of database systems by ramez elmarsi and shamkant b.navathe

2. OVERALL DESCRIPTION

2.1 PRODUCT PERSPECTIVE

A distributed airline database system stores the following information.

- **Flight details:**
  It includes the originating flight terminal and destination terminal, along with the stops in between, the number of seats booked/available seats between two destinations etc.
- **Customer description:**
  It includes customer code, name, address and phone number. This information may be used for keeping the records of the customer for any emergency or for any other kind of information.
- **Reservation description:**
  It includes customer details, code number, flight number, date of booking, date of travel.

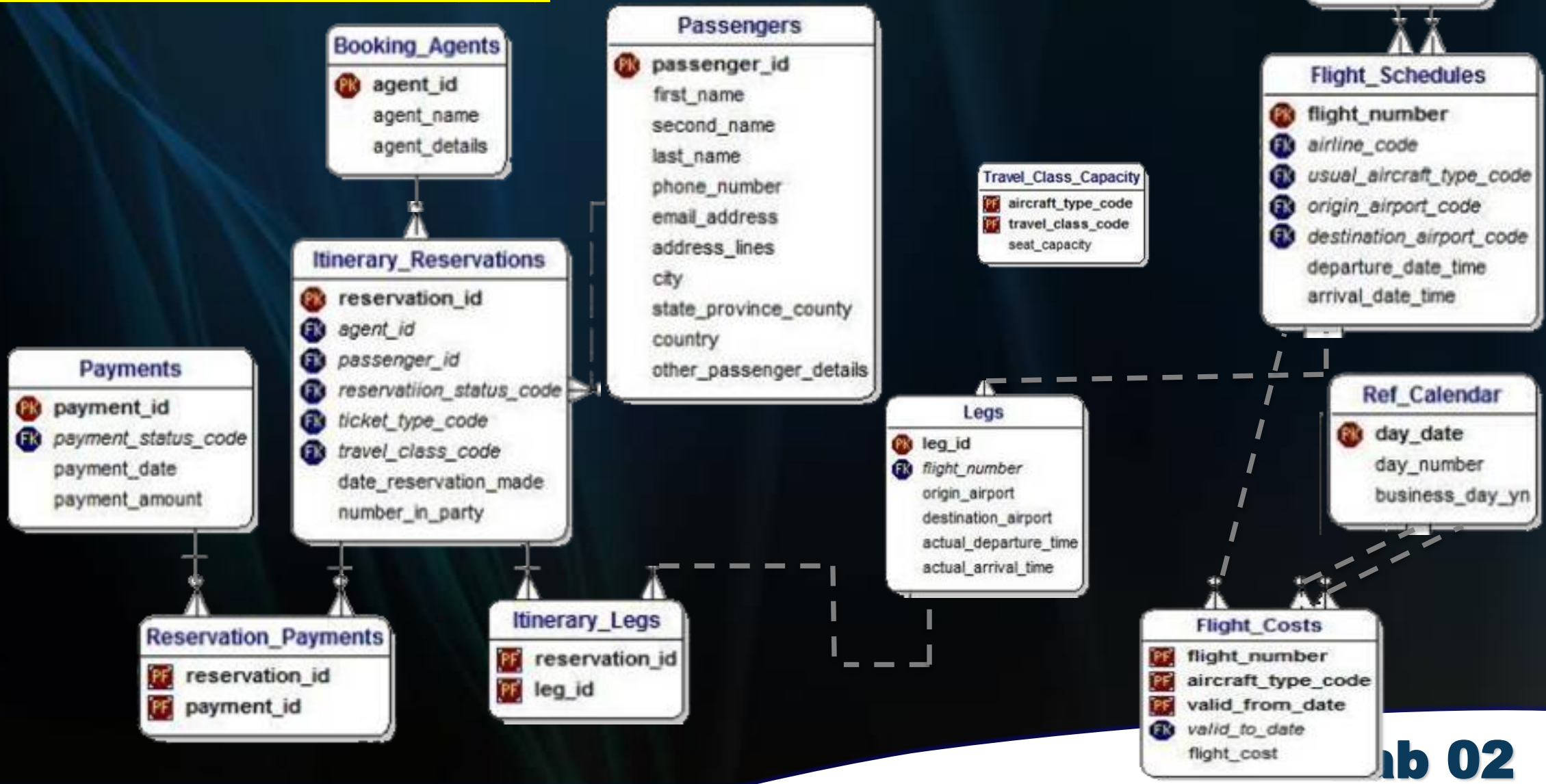
2. OVERALL DESCRIPTION(cont.)

2.2 PRODUCT FEATURES

The major features of airline database system as shown in below entity–relationship model (ER model)

SRS

2. OVERALL DESCRIPTION (ERD)



2. OVERALL DESCRIPTION(cont.)

2.3 USER CLASSIFICATION (ROLES)

Users of the system should be able to retrieve flight information between two given cities with the given date/time of travel from the database.

A route from city A to city B is a sequence of connecting flights from A to B such that:

a) There are at most two connecting stops, excluding the starting city and destination city of the trip,

b) The connecting time is between one to two hours. The system will support two types of user privileges, Customer, and Employee. Customers will have access to customer functions, and the employees will have access to both customer and flight management functions. The customer should be able to do the following functions:

- ❑ Make a new reservation
  - One-way
  - Round-Trip
  - Multi-city
  - Flexible Date/time
  - Confirmation
- ❑ Cancel an existing reservation
- ❑ View his itinerary

2. OVERALL DESCRIPTION(cont.)

The employee should have following management functionalities:

CUSTOMER FUNCTIONS.

- Get all customers who have seats reserved on a given flight.
- Get all flights for a given airport.
- View flight schedule.
- Get all flights whose arrival and departure times are on time/delayed.
- Calculate total sales for a given flight.

ADMINISTRATIVE

- Add/Delete a flight
- Add a new airport
- Update fare for flights.
- Add a new flight leg instance.
- Update departure/arrival times for flight leg instances.

Each flight has a limited number of available seats. There are a number of flights which depart from or arrive at different cities on different dates and time.

2. OVERALL DESCRIPTION(cont.)

2.4 OPERATING ENVIRONMENT

Operating environment for the airline management system is as listed below. Include the details as per your application.

- distributed database
- client/server system
- Operating system: Windows.
- database: sql+ database
- platform: vb.net/Java/PHP

2.5 DESIGN and IMPLEMENTATION CONSTRAINTS

1. The global schema, fragmentation schema, and allocation schema.
2. SQL commands for above queries/applications
3. How the response for application 1 and 2 will be generated. Assuming these are global queries. Explain how various fragments will be combined to do so.
4. Implement the database at least using a centralized database management system.

2. OVERALL DESCRIPTION(cont.)

2.6 ASSUMPTION DEPENDENCIES

Let us assume that this is a distributed airline management system and it is used in the following application:

- A request for booking/cancellation of a flight from any source to any destination, giving connected flights in case no direct flight between the specified Source-Destination pair exist.
- Calculation of high fliers (most frequent fliers) and calculating appropriate reward points for these fliers.

Assuming both the transactions are single transactions, we have designed a distributed database that is geographically dispersed at four cities Delhi, Mumbai, Chennai, and Kolkatta as shown in fig. below.

3. Functional Requirements

3.1 DISTRIBUTED DATABASE:

Distributed database implies that a single application should be able to operate transparently on data that is spread across a variety of different databases and connected by a communication network

3.2 CLIENT/SERVER SYSTEM

The term client/server refers primarily to an architecture or logical division of responsibilities, the client is the application (also known as the front-end), and the server is the DBMS (also known as the back-end).

A client/server system is a distributed system in which,

- ❑ Some sites are client sites and others are server sites.
- ❑ All the data resides at the server sites.
- ❑ All applications execute at the client sites.

4. NONFUNCTIONAL REQUIREMENT

4.1 PERFORMANCE REQUIREMENTS

In order to assess the performance of a system the following must be clearly specified:

• Response Time: response time would be improved after using a normalized database.

• Workload: Multithreaded environment would be programmed and for that hardware has been procured by the company.

• Scalability: data would increase on daily basis so extensive hardware and latest technology would not impact on software efficiency.

4.2 SAFETY REQUIREMENTS

If there is extensive damage to a wide portion of the database due to catastrophic failure, such as a disk crash, the recovery method restores a past copy of the database that was backed up to archival storage (typically tape) and reconstructs a more current state by reapplying or redoing the operations of committed transactions from the backed up log, up to the time of failure.

4.3 SECURITY REQUIREMENTS

Security systems need database storage just like many other applications. However, the special requirements of the security market mean that vendors must choose their database partner carefully.

4. NONFUNCTIONAL REQUIREMENT(cont.)

4.4 SOFTWARE QUALITY ATTRIBUTES

AVAILABILITY: The flight should be available on the specified date and specified time as many customers are doing advance reservations.

CORRECTNESS: The flight should reach start from correct start terminal and should reach the correct destination.

MAINTAINABILITY: The administrators and flight incharges should maintain correct schedules of flights.

USABILITY: The flight schedules should satisfy a maximum number of customers needs

Functional Non functional summary for our understanding

functional requirements: A functional requirement describes what a software system should do, the way it functions. E.g. for security authentication module is made, for user-friendliness every user will have its desired dashboard, for quick responsive database have smart indexing techniques implemented.

Non-functional requirements are not straight forward requirement of the system rather it is related to usability & deals with constraints on how the system will do so. E.g. must be secured ,user-friendly and quick responsive.

5. INTERFACE REQUIREMENTS

5.1 USER INTERFACES

Front-end software: Vb.net version
Back-end software: SQL+

5.2 HARDWARE INTERFACES

Since neither the mobile application nor the web portal have any designated hardware, it does not have any direct hardware interfaces. The physical GPS is managed by the GPS application in the mobile phone and the hardware connection to the database server is managed by the underlying operating system on the mobile phone and the web server.

5.3 SOFTWARE INTERFACES

Following are the software used for the flight management online application. Include the software details as per your project:

Software used	Description
Operating system	We have chosen Windows operating system for its best support and user-friendliness.
Database	To save the flight records, passengers records we have chosen SQL+ database.
VB.Net	To implement the project we have chosen Vb.Net language for its more interactive support.

5. INTERFACE REQUIREMENTS

5.4 COMMUNICATION INTERFACES

This project supports all types of web browsers. We are using simple electronic forms for the reservation forms, ticket booking etc.

The communication between the different parts of the system is important since they depend on each other. However, in what way the communication is achieved is not important for the system and is therefore handled by the underlying operating systems for both the mobile application and the web portal.

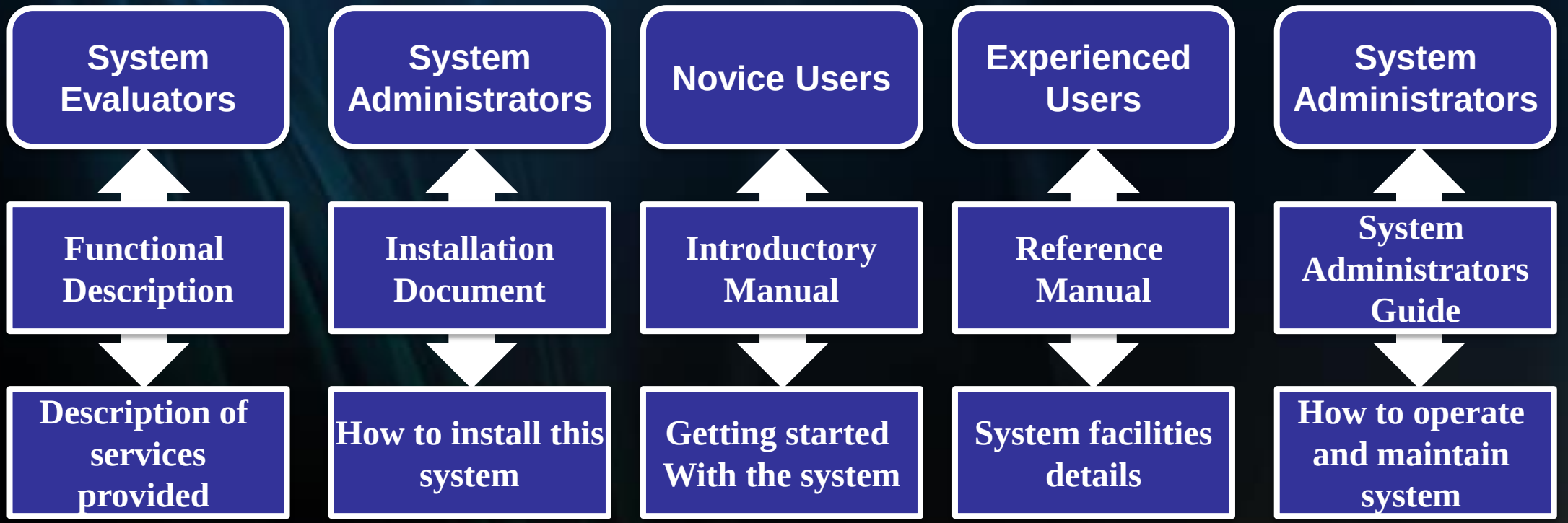
Process Documentation

- Process documents record the process of development and maintenance.
- Plans, schedules, process quality documents and organizational and project standards are process documentation.
- The major characteristic of process documentation is that most of it becomes outdated.
- Plans may be drawn up on a weekly, fortnightly or monthly basis.
- Progress will normally be reported weekly.
- Memos record thoughts, ideas and intentions, which changes a lot.

Product Documentation

Product documentation includes :

- user documentation, which tells users how to use the software product accordingly
- system documentation, which is principally intended for maintenance engineers. Let see different types of user documents



Documentation Standards

Documents produced according to appropriate standards have a consistent appearance, structure and quality. The standards that may be used in the documentation process are:

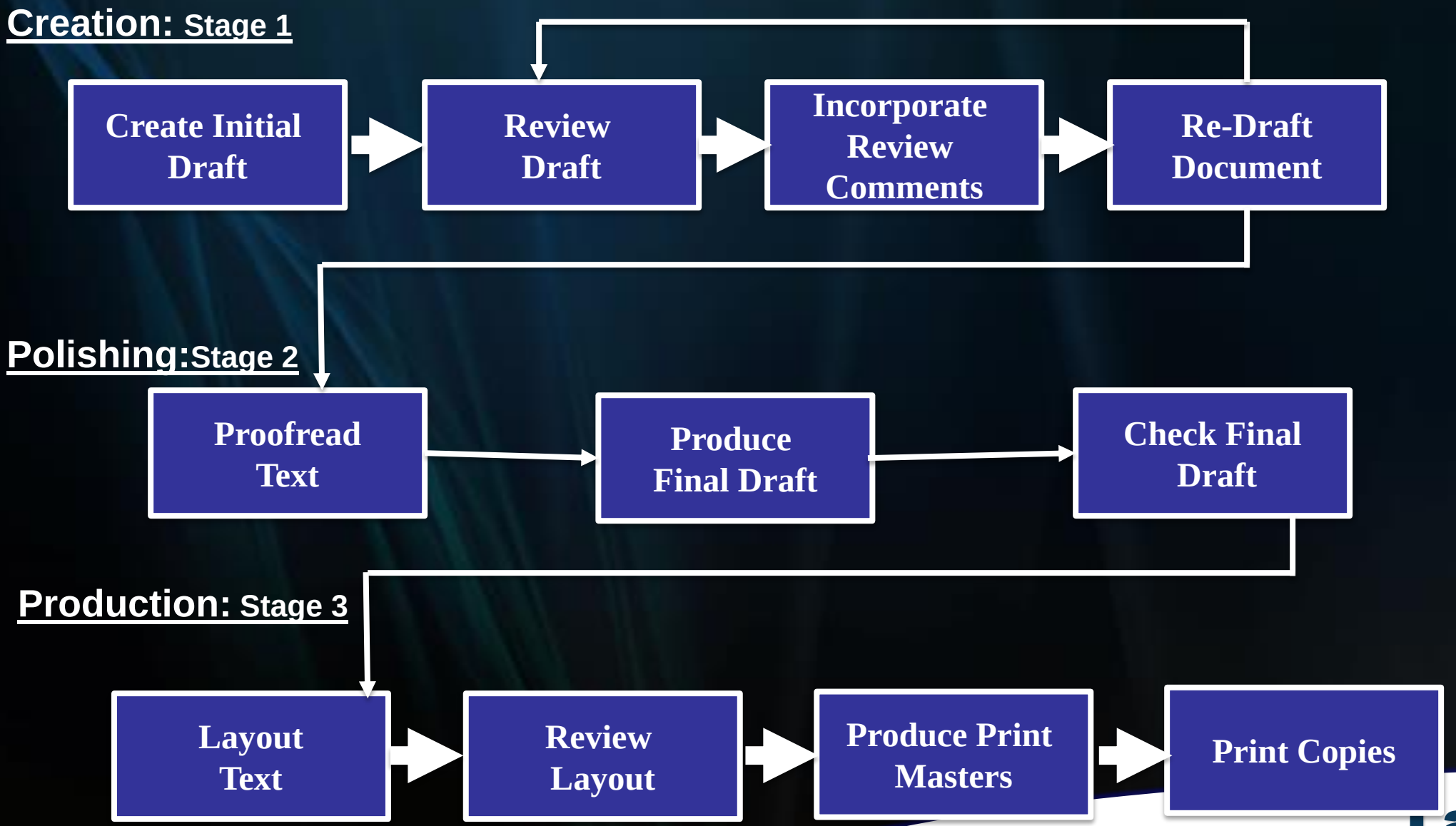
1. ***Process standards*** benchmark steps to create a document
2. ***Product standards*** benchmark outcomes of a document
3. ***Interchange standards :*** It is important to exchange copies of documents via electronic mail and to store documents in databases. Interchange standards ensure that all electronic copies of documents are compatible.

Writing style

Technical writing is a skill rather than a science but some broad guide-lines about how to write well are:

1. Use active rather than passive tenses
2. Use grammatically correct constructs and correct spelling
3. Be Precise
6. Make use of headings and sub-headings.
7. Itemize facts wherever possible

Stages of document preparation



Thanks