BASIC TEXTILE OPERATION

NTQF Level -1

Learning Guide -50

Unit of Competence: Prepare Yarn for Weaving and Knitting

Module Title: Preparing Yarn for Weaving and Knitting

LG Code: IND BTO1 M13 LO4-LG-50

TTLM Code: IND BTO1 TTLM 0919v1

LO-4: Complete records

Instruction Sheet	Learning Guide #50

This learning guide is developed to provide you the necessary information regarding the following **content coverage** and topics:

- Recording production information.
- Recording faults, break downs and other documentations

This guide will also assist you to attain the learning outcome stated in the cover page. Specifically, upon completion of this Learning Guide, **you will be able to:**

- ❖ Production information is recorded as per the format.
- ❖ Faults, break downs and other documentations are recorded as per the form

Learning Instructions:

1. Read the specific objectives of this Learning Guide.
2. Follow the instructions described in number 3 to 9.
3. Read the information written in the "Information Sheets 1". Try to understand what are being discussed. Ask you teacher for assistance if you have hard time understanding them.
4. Accomplish the "Self-check 1".
5. Ask from your teacher the key to correction (key answers) or you can request your teacher to correct your work. (You are to get the key answer only after you finished answering the Self-check 1).
6. If you earned a satisfactory evaluation proceed to "Information Sheet 2". However, if your rating is unsatisfactory, see your teacher for further instructions.
7. Read the information written in the "Information Sheet 2". Try to understand what are being discussed. Ask you teacher for assistance if you have hard time understanding them.
8. Accomplish the "Self-check 2".
9. Ask from your teacher the key to correction (key answers) or you can request your teacher to correct your work. (You are to get the key answer only after you finished answering the Self-check 2).

Information Sheet-1	Recording production information

After yarn preparation is completed in each batch/cycle, the operator has responsibility of recording production information. Production information may include: yarn batch, Date of yarn preparation, yarn count, type of yarn and their respective amount also.

Production information also may include:

- Drawings (location drawings, component drawings and dimensioned diagrams).
- Specifications, design criteria and calculations (specification information can be included on drawings or in a separate specification, but information should not be duplicated as this can become contradictory and may cause confusion).
- Bills of quantities or schedules of work (schedules of work are 'without quantities' instructional specifications often produced by designers on smaller projects for pricing, or for items such as builders work and fixing schedules, such as sanitary fittings, doors, windows, ironmongery, light fittings, louvers, roller shutters, diffusers, grilles and manholes).
- There should be a particular emphasis on equipment with long manufacturing times, such as switchgear, chiller units, lifts, escalators or bespoke cladding systems, and on front-end construction such as service diversions, demolition, setting out details, underground drainage, piling and ground works.
- Definitions and rules relating to drawn information for 'with quantities' projects are described the New Rules of Measurement. See New Rules of Measurement for more information.
- Increasingly, software is used to prepare elements of production information such as computer aided design (CAD) to prepare drawings, common data environments (CDE), and proprietary systems for the preparation of specifications.
- The advent of building information modelling (BIM) can allow the automatic generation of all elements of production information from a single coordinated model, resulting in a reduction in errors and so costs.

Self-Check – 1	**Written Test**

Directions: Answer all the questions listed below. Use the Answer sheet provided in the next page

1. Discuss about recording production information. (4 point)

Note: **Satisfactory rating – 4 points** **Unsatisfactory - below 4 points**

Answer Sheet

Score = ____________

Rating: _____________

Name: ___________________________ Date: ________________

1. ____

____

____

____

____

____

____

____

____

____

____

____

Information Sheet-2	**Recording faults, break downs and other documentations**

2.1. Recording faults

After work is completed, not only production information recorded, but also any yarn faults, break-down or other must be recorded on separate format to find and fix solutions. Yarn faults may include: thin place, thick place, wrong yarn count, yarn contamination and etc.

A **fault** occurs when a human error results in a mistake in some software product. That is, the fault is the encoding of the human error. For example, a developer might misunderstand a user interface requirement, and therefore create a design that includes the misunderstanding. The design fault can also result in incorrect code, as well as incorrect instructions in the user manual. Thus, a single error can result in one or more faults, and a fault can reside in any of the products of development.

On the other hand, a **failure** is the departure of a system from its required behavior. Failures can be discovered both before and after system delivery, as they can occur in testing as well as in operation. It is important to note that we are comparing actual system behavior with *required* behavior, rather than with *specified* behavior, because faults in the requirements documents can result in failures, too.

During both test and operation, we observe the behavior of the system. When undesirable or unexpected behavior occurs, we report it as an **incident**, rather than as a failure, until we can determine its true relationship to required behavior. For example, some reported incidents may be due not to system design or coding faults but instead to hardware failure, operator error or some other cause consistent with requirements. For this reason, our approach to data collection deals with incidents, rather than failures.

The reliability of a software system is defined in terms of incidents observed during operation, rather than in terms of faults; usually, we can infer little about reliability from fault information alone. Thus, the distinction between incidents and faults is very important. Systems containing many faults may be very reliable, because the conditions that trigger the faults may be very rare. Unfortunately, the relationship between faults and incidents is poorly understood; it is the subject of a great deal of software engineering research.

One of the problems with problems is that the terminology is not uniform. If an organization measures its software quality in terms of faults per thousand lines of code, it may be impossible to compare the result with the competition if the meaning of "fault" is not the same. The software engineering literature is rife with differing meanings for the same terms. Below are just a few examples of how researchers and practitioners differ in their usage of terminology.

To many organizations, *errors* often mean faults. There is also a separate notion of "processing error," which can be thought of as the system state that results when a fault is triggered but before a failure occurs. This particular notion of error is highly relevant for software fault tolerance (which is concerned with how to prevent failures in the presence of processing errors).

Anomalies usually mean a class of faults that are unlikely to cause failures in themselves but may nevertheless eventually cause failures indirectly. In this sense, an anomaly is a deviation from the usual, but it is not necessary wrong. For example, deviations from accepted standards of good programming practice (such as use of non-meaningful names) are often regarded as anomalies.

Defects normally refer collectively to faults and failures. However, sometimes a defect is a particular class of fault. For example, Mellor uses "defect" to refer to faults introduced prior to coding.

Bugs refer to faults occurring in the code.

Crashes are a special type of incident, where the system ceases to function.

Until terminology is the same, it is important to define terms clearly, so that they are understood by all who must supply, collect, analyze and use the data. Often, differences of meaning are acceptable, as long as the data can be translated from one framework to another.

We also need a good, clear way of describing what we do in reaction to problems. For example, if an investigation of an incident results in the detection of a fault, then we make a change to the product to remove it. A change can also be made if a fault is detected during a review or inspection process. In fact, one fault can result in multiple changes to one product (such as changing several sections of a piece of code) or multiple changes to multiple products (such as a change to requirements, design, code and test plans).

We describe the observations of development, testing, system operation and maintenance problems in terms of incidents, faults and changes. Whenever a problem is observed, we want to record its key elements, so that we can then investigate causes and cures. In particular, we want to know the following:

1. *Location*: Where did the problem occur?
2. *Timing*: When did it occur?
3. *Mode*: What was observed?
4. *Effect*: Which consequences resulted?
5. *Mechanism*: How did it occur?
6. *Cause*: Why did it occur?
7. *Severity:* How much was the user affected?
8. *Cost:* How much did it cost?

Fault Report

Location: within-system identifier, such as module or document name. The IEEE Standard Classification for Software Anomalies, provides a high-level classification that can be used to report on location.

Timing: phases of development during which fault was created, detected and corrected. Clearly, this part of the fault report will need revision as a causal analysis is performed. It is also useful to record the time taken to detect and correct the fault, so that product maintainability can be assessed.

Mode: type of error message reported, or activity which revealed fault (such as review). The Mode classifies what is observed during diagnosis or inspection. The IEEE standard on software anomalies, provides a useful and extensive classification that we can use for reporting the mode.

Effect: failure caused by the fault. If separate failure or incident reports are maintained, then this entry should contain a cross-reference to the appropriate failure or incident reports. **Mechanism:** how source was created, detected, corrected. Creation explains the type of activity that was being carried out when the fault was created (for example, specification, coding, design, and maintenance). Detection classifies the means by which the fault was found (for example, inspection, unit testing, system testing, integration testing), and correction refers to the steps taken to remove the fault or prevent the fault from causing failures.

Cause: type of human error that led to fault. Although difficult to determine in practice, the cause may be described using a classification suggested by Collofello and Balcom: [Collofello and Balcom 1985]: a) communication: imperfect transfer of information; b) conceptual: misunderstanding; or c) clerical: typographical or editing errors

Severity: refer to severity of resulting or potential failure. That is, severity examines whether the fault can actually be evidenced as a failure, and the degree to which that failure would affect the user.

Cost: time or effort to locate and correct; can include analysis of cost had fault been identified during an earlier activity

Changes

Once a failure is experienced and its cause determined, the problem is fixed through one or more changes. These changes may include modifications to any or all of the development products, including the specification, design, code, test plans, test data and documentation. Change reports are used to record the changes and track the products most affected by them. For this reason, change reports are very useful for evaluating the most fault-prone modules, as well as other development products with unusual numbers of defects. A typical change report may look like this:

Change Report

- ✓ **Location:** identifier of document or module affected by a given change.
- ✓ **Timing:** when change was made
- ✓ **Mode:** type of change
- ✓ **Effect:** success of change, as evidenced by regression or other testing
- ✓ **Mechanism:** how and by whom change was performed
- ✓ **Cause:** corrective, adaptive, preventive or perfective
- ✓ **Severity:** impact on rest of system, sometimes as indicated by an ordinal scale
- ✓ **Cost:** time and effort for change implementation and test

2.2. Recording break downs and other documentations

The »Maintenance Documentation describes all measures required in order to ensure and maintain the functional capability of the system. Maintenance is scheduled and executed at certain intervals, in case of a vehicle, for example, annually or every 15,000 km. The maintenance documentation is intended for persons planning and executing the maintenance procedures

Maintenance Plan

The »Maintenance Plan describes the individual maintenance measures and their schedule. The maintenance measures may be summarized in maintenance levels. Maintenance procedures may be executed during operation or if the operation is interrupted.

The maintenance plan may also include the equipment maintenance record if an individual maintenance plan exists for every system. If this is not the case, the equipment maintenance record shall be conducted in suitable form, e.g., as service manual, maintenance manual or material history report.

A maintenance plan may include the following:

- Warning statements and notes of caution
- Serial number
- System element or test location of the measure, possibly with item number of the spare parts catalog
- Standard/special tools, measuring and test equipment
- Procedure to be executed
- Statement on admissible wear-and-tear parts, operating fluids and tolerances
- Work schedule, depending on operating parameters (e.g., hours, time, number and type of use)

Maintenance Instructions

The »Maintenance Instructions describes the execution of the different maintenance measures in repeatable procedural steps. The maintenance instruction will only be prepared for measures which require additional explanations not included in the »Maintenance Plan. The disposal of wear-and-tear parts and operating fluids must be considered. The use of measuring and test equipment and necessary tools will be explained.

The maintenance instruction may include, but not be limited to, the following:

- Warning statements and notes of caution

- Cleaning of the system
- Standard/special tools, measuring and test equipment
- Replacement of wear-and-tear parts and operating fluids
- Monitoring of operating parameters

Self-Check -2	**Written Test**

Directions: Answer all the questions listed below. Use the Answer sheet provided in the next page:

1. Discuss fault recording.(2 point)
2. Discuss break downs and other documentations. (2 point)

***Note:* Satisfactory rating - 4 points** **Unsatisfactory - below 4 points**

Answer Sheet

Score = ___________

Rating: ____________

Name: _________________________ Date: _______________

Short Answer Questions

1. ____

____

____

____

2. ____

____

____

____

____

____

____

____

List of Reference Materials

1- Wynne, A. The Motivate Series, Textiles, 1997.

2- Handbook of technical textiles edited by a r horrocks and s c anand

3- Textile Sizing (Goswami-2004), by BHUVENESH C. GOSWAMI Clemson University Clemson, South Carolina, U.S.A.

4- Textile dictionary by Celanese Acetate Three Park Avenue New York, NY 10016

5- Reference books of textile technologies By Giovanni Castelli, Salvatore Maietta, Giuseppe Sigrisi, Ivo Matteo Slaviero

6- Cotton: Science and technology Edited by S. Gordon and Y-L. Hsieh