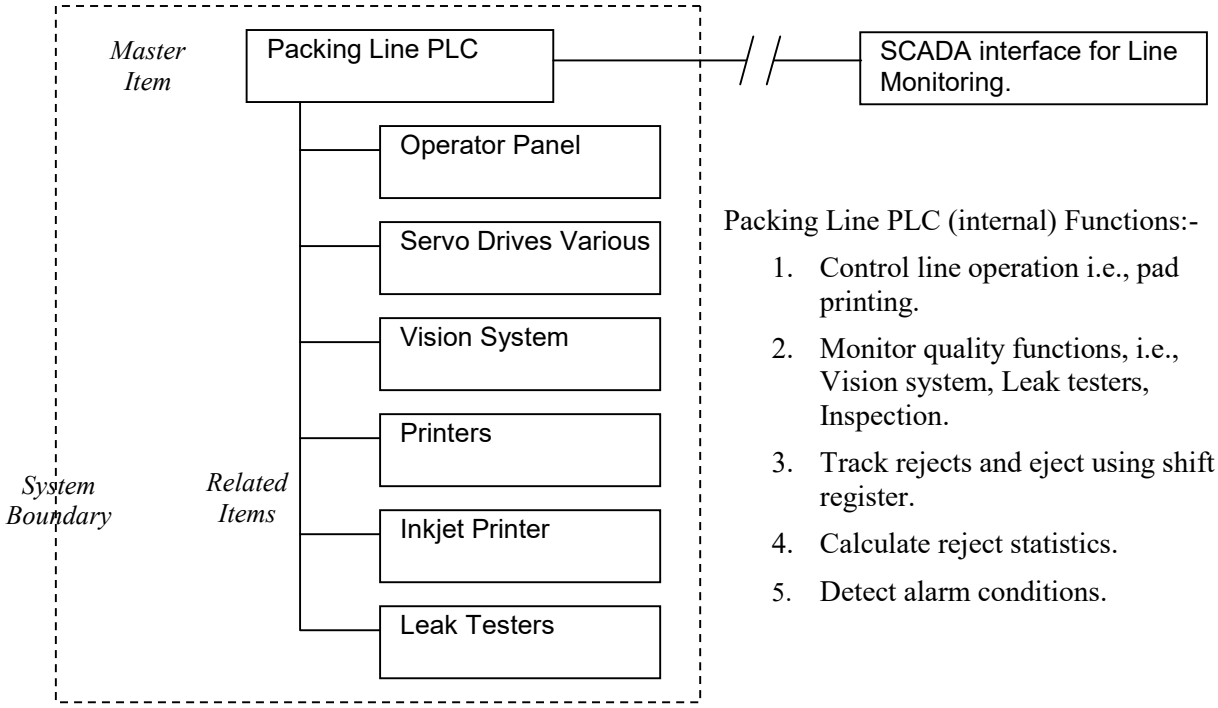
An illustration of the structure of a Computerised System is the following Finishing Line PLC:



Packing Line PLC (internal) Functions:-

1. Control line operation i.e., pad printing.
2. Monitor quality functions, i.e., Vision system, Leak testers, Inspection.
3. Track rejects and eject using shift register.
4. Calculate reject statistics.
5. Detect alarm conditions.

3. Criticality Assessment (Quality Impact)

Validation is concerned with assuring that our manufacturing processes, activities and systems deliver product that reliably meets quality standards. Product Quality attributes are Identity, Safety, Efficacy, Purity, and Evidence. Examples of quality characteristics are shown in the table in Section 10. Consideration of the significance of electronically records generated may influence the Criticality Assessment, as outlined in Section 11.

3.1. Direct Impact

Impact Assessments look individually at the Items within Computerised Systems to evaluate the affect of their Functions on product quality.

The following list of questions assess whether an Item/Function has a "Direct Impact" on the quality of a product/process or integrity of stored data.
(If the answer to any question is "YES" the Item/Function has Direct Impact.)

The Criteria below should be used to assist in formulating a judgement based on the comprehensive understanding of the product, process and the nature of the system. They should NOT be used to replace the exercise of professional judgment by appropriately qualified personnel.

1. Is the Item used to demonstrate compliance with the registered process?
2. Will the normal operation or control by the Item have a direct effect on the product/process quality, (including ingredients and product components)?
3. Will failure of the Item or its alarms have a direct effect on product quality or efficacy?
4. Is information from this Item recorded as part of batch record, lot release data, or other GMP related documentation?
5. Does the Item control critical process functions that may affect product quality (and there is

to lower levels should be assessed on the basis of their GAMP rating however their interaction with higher levels within the Item must also be considered.

Note that the Data level has no GAMP rating. If this level however, has a GMP role it may determine the overall Impact of the system (i.e. Direct). Examples of ratings for various types of records are shown in Section 11. Data layers with GMP Impact require control measures to preserve their documentation attributes (i.e. accuracy, authenticity, availability and integrity). These control measures should be recorded on the Impact Assessment form and included in the Master Validation Plan.

5. Overall Risk-Profile Classification

Combine the Impact rating and GAMP category to determine a Validation strategy (C number):

	Impact Assessment (Criticality)		
	No Impact on GXP Functions No impact on the performance or operation of GxP Functions	**Indirect Impact on GXP Functions** Items that may affect the performance or operation of other Items which have Direct Impact on GxP Functions	**Direct Impact on GXP Functions** Items that have a direct effect on the performance or operation of GxP Functions
GAMP Category (Complexity)			
1. Operating systems	**C1 Validation:** Record Version & GEP Functional	**C1 Validation:** Record Version & GEP Functional	**C1 Validation:** Record Version & GEP Functional
2. Firmware (Instruments and controllers)	**C1 Validation:** Record Version & GEP Functional	**C1 Validation:** Record Version & GEP Functional	**C2 Validation:** Record Configuration and Version No & GEP Functional
3. Standard packages	**C1 Validation:** Record Version & GEP Functional	**C2 Validation:** Record Configuration and Version No & GEP Functional	**C3 Validation:** **Minimal Functional**
4. Configurable packages	**C2 Validation:** Record Configuration and Version No & GEP Functional	**C3 Validation:** **Minimal Functional**	**C4 Validation:** **Some Functional** **Minimal Structural**
5. Custom-built	**C3 Validation:** **Minimal Functional**	**C4 Validation:** **Some Functional** **Minimal Structural**	**C5 Validation:** **Extensive Functional** **Extensive Structural**

the site can be reduced. Tests recorded on protocols developed by suppliers do not require copying to site formats (so long as the content is appropriate). Conversely, where there is reason to be concerned about the assurance provided by a supplier, additional testing and input by the site may be necessary. For maximum benefit, any extra involvement from the site should be provided as early as possible within the Development Lifecycle.

- Other Compliance Requirements. Where assurance of software performance is required for other reasons (e.g. compliance with EHS regulations) additional testing might be considered. Such testing may utilise the formats and structures of Validation protocols, as appropriate.

8. Qualification

As shown in Figure 6.1, the overall qualification of the Computerised System would be comprised of the activities of validation corresponding to the categories of the **individual** Items, once the criticality and the complexity of the Items are established (refer to section 5.0).

8.1. Good Engineering Practices

All systems, regardless of their Quality Impact, are to be supplied and developed in accordance with Good Engineering Practices (GEP). For many systems there will be no separate requirement for Validation and GEP alone is sufficient. Evidence of GEP includes:

- Developments are designed or specified against agreed requirements
- Competent personnel (including contractors) are selected for the task
- Full consideration is given to EHS, Operating, Maintenance and Standards requirements
- Completed works are inspected, tested, commissioned and recorded appropriately.

The “GEP-alone” approach does not imply an absence of documentation; rather there is a reduced need for review and approval of this documentation by Quality Assurance personnel.

8.2. Structural Verification

Structural verification involves inspection and assessment of the actual source code by a suitably qualified person (who is not the programmer). Documentation used to support verification include logic diagrams, description of modules, definitions of all variables and specification of all inputs and outputs. Structural verification is used to assess:

1. General application of good programming standards and practices. Some programming practices are known to be associated with operational failures, or to hamper ongoing maintenance. Depending on the software being programmed, the following are examples of issues that might be verified:

- code layout is logical and the flow is easy to follow, including adequate comments that aid understanding by others who may need to maintain it in the future,
- dead code and open-loops have been removed or commented out,
- variables are sensibly named,
- confounding special values are excluded e.g. divide by zero or square root of a negative number,
- operations with invalid or missing data are prevented (e.g. by checking for empty strings, correct data type, values within limited range),
- parameters are initialised to a known value prior to use to prevent unexpected results,
- the program operates safely when abnormal (error) conditions occur,
- invalid, illegal or adverse conditions such as alarms, alerts, errors and hardware failures, are identified and highlighted to the user in a way that allows appropriate response,
- databases and fields are correctly indexed,

