RISK-BASED PROGRAM VALIDATION FOR CLINICAL TRIAL ANALYSIS AND REPORTING
Objective

• Propose methodology for clinical programmers to align validation of clinical programs with regulations
  – Implement “improved and more efficient approaches to clinical trial design, conduct, oversight, recording and reporting while continuing to ensure human subject protection and data integrity.”

• Suggest a way to adopt a risk-based approach to validation allowing us to
  – Vary degree of testing rigor
  – Focus validation activities on the most critical analyses
  – Work efficiently while still ensuring high quality, accurate analysis of data.
  – Accurately analyze the safety and efficacy of medications
  – Ensure human subject protection.
Regulations

• Guidances to identify, assess, control, communicate, review and report risks as they relate to program validation for clinical trial analyses.

• ICH E6 (R2) Good Clinical Practice: Integrated Addendum to ICH E6(R1)

• ICH Quality Risk Management (Q9)
What is Risk

In clinical trials, risk can be defined as “the combination of the probability of occurrence of harm and the severity of that harm.”
Principles of Quality Risk Management

- Level of risk to quality and accuracy of our results
- Protection of the patient linked to that accuracy
- Level of effort that aligns with the risk.
ICH – Q9 Quality Management Plan

Quality Management Plan in Q9 can be adapted to fit program validation:

• Determine timing, outputs/deliverables and decision-making responsibilities

• Statistician, clinician and clinical programmer partner to follow the risk management process and evaluate risks

• When beginning a quality risk management process for clinical programming, the team must work to identify potential risks associated with each program including the potential harm or human health impact of the associated analyses. A leader and resources to support the process must be identified.
ICH Q9 – Figure 1 – Example of a typical quality risk management process
Suggested quality risk management process for program validation

1. **Assess Risk**
   - Determine criticality of each program
   - Assign risk category

2. **Control Risk**
   - Document validation plan to align with risk category
   - Validation Plan

3. **Review Risk**
   - Review risks/exceptions

4. **Communicate Risk**

   - Risk Management Tools
   - Unacceptable
Assess Risk

- Assess Risk: Determine criticality of each program, Assign risk category
- Control Risk: Document validation plan to align with risk category
- Communicate Risk
- Review Risk: Review risks/exceptions

Risk Management Tools

Unacceptable
## Risk Category – Probability and Severity (High, Medium or Low)

<table>
<thead>
<tr>
<th>Risk Severity</th>
<th>Risk Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>High:</strong> Critical analysis, primary hypotheses for efficacy or safety, CSR/CTD analyses, cross projects</td>
<td>High: New logic, Complex algorithm, Use across areas, Major changes to critical derivations</td>
</tr>
<tr>
<td><strong>Medium:</strong> Appendices, 2ndary/Exploratory analyses, Publications</td>
<td>High</td>
</tr>
<tr>
<td><strong>Low:</strong> Non-submitted analyses, Cosmetic changes (labels, variable names, footnotes)</td>
<td>Medium</td>
</tr>
</tbody>
</table>
Control Risk

- Assess Risk
  - Determine criticality of each program
  - Assign risk category

- Control Risk
  - Document validation plan to align with risk category

- Communicate Risk

- Review Risk
  - Review risks/exceptions

- Risk Management Tools

Unacceptable
Validation Activities – based on Risk Category

<table>
<thead>
<tr>
<th>Category</th>
<th>Required Validation/QC</th>
</tr>
</thead>
</table>
| High     | • Developer testing activities  
             • Statistical review of output with focus on complex derivations  
             • Double independent coding of critical dataset/variables (list variables in plan)  
             • User acceptance testing across multiple trials as appropriate  
             • Clinical review of dry run TLFs  
             • Independent code review |
| Medium   | • Developer testing activities  
             • Statistical review of output  
             • Clinical review of dry run TLFs  
             • Independent code review |
| Low      | • QC of logs and output  
             • Customer review of dry run TLFs  
             • Cross table count checks  
             • Compare to prior output for changes |
Validation Plan

1. Assess Risk
   - Determine criticality of each program
   - Assign risk category

2. Control Risk
   - Document validation plan to align with risk category

3. Review Risk
   - Review risks/exceptions

START

Validation Plan

Risk Management Tools

Unacceptable
Review Risk

1. Assess Risk
   - Determine criticality of each program
   - Assign risk category

2. Control Risk
   - Document validation plan to align with risk category

3. Communicate Risk

4. Risk Management Tools

5. Validation Plan

6. Review Risk
   - Review risks/exceptions

START

Unacceptable
Communicate Risk

**Assess Risk**
- Determine criticality of each program
- Assign risk category

**Control Risk**
- Document validation plan to align with risk category

**Review Risk**
- Review risks/exceptions

**Validation Plan**

**Risk Management Tools**

Unacceptable
Risk Management Tools

START

Assess Risk
Determine criticality of each program
Assign risk category

Control Risk
Document validation plan to align with risk category

Review Risk
Review risks/exceptions

Risk Management Tools

Communicate Risk

Validation Plan

Unacceptable
Conclusion

• Consider having a quality risk management plan for validation of programs

• Create standard risk-based program categories and define expected validation activities aligned with each
  • Testing activities align with the risk associated with results analysis
THANKS

Your comments and questions are valued and encouraged.

Jane Marrer
Merck Sharp & Dohme Corp
Email: jane_marrer@merck.com