CDISC ADaM specifications documentation is the core of the programming activities in FDA submission. It serves as the primary source for ADaM programming and validation, define.xml and reviewer guide, and also helps in establishing metadata traceability. It is preferable to create ADaM specifications in Word® format to facilitate both reviewing and approving of specifications documentation by statisticians and/or validators. Since derivation rules may be complex and subject to constant change during the whole ADaM programming activities, it is desirable to automatically keep track of different versions of ADaM programming specifications. It is more beneficiary for sponsors to keep track of different versions when ADaM programming is outsourced to external vendors. This paper introduces an SAS macro to automatically detect and report any revisions between the new version and the old versions of programming specifications. It helps to generate necessary documents for version control, establishes traceability and achieves high efficiency in FDA submission.

**Goal:**
- Track any changes from any old versions of specifications
- Help to finalize the programming specification at the very early stage in an efficient way
- Ensure the traceability and clear documentation for FDA submission
- Achieve Cost-Effectiveness and Efficiency

**Introduction: Programming Specification for ADaM Dataset(s)**

1.1.1 **ADaM: Subject Level Analysis Dataset**

<table>
<thead>
<tr>
<th>Domain Information Table</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dataset</td>
</tr>
<tr>
<td>Program Name</td>
</tr>
<tr>
<td>Description</td>
</tr>
<tr>
<td>Additional Information Variables</td>
</tr>
<tr>
<td>Structure</td>
</tr>
<tr>
<td>Metadata</td>
</tr>
<tr>
<td>Notes</td>
</tr>
</tbody>
</table>

**Transfer from Word to SAS Format**

**Version Control - Macro %read_spec**

```
%macro read_spec(indir=,specsnm=,outdir=,newdtnm=,runorder=);
```

Where
- **INDIR:** Full Path for ADaM programming specification.
- **SPECSNM:** Name of ADaM programming specification.
- **OUTDIR:** Full Path for output SAS dataset which contains attributes of ADaM variables.
- **NEWDTNM:** A valid SAS dataset name for SAS dataset containing current specifications information.
- **RUNORDER:** A valid numerical, defining the order for a specific domain to run (in the final run).

**Function:**
- Read information from individual ADaM specifications in CSV format
- Store both word version and SAS creation of the programming specifications with time stamp in a study subfolder, named as ‘history’.

**Sample Code:**
```
** Get date of programming running as time stamp ***;
*** data _null_;
*** day = compress(put(today(),yymmd10.));
*** call symput('day',strip(day));
*** run;
*** Output a SAS dataset of specification with time stamp to a subfolder ‘history’ ***;
*** data.newdtnm._.day.;
*** set __vars;
*** Copy a Word Version with time stamp to a subfolder ‘history’ ***;
*** options nomain NOXSYNC;
```

**Conclusion**
- The macro makes it possible to track any changes of the specifications in an efficient way.
- The traceability can be achieved with the storage of the previous version of specifications.
- The documentation for audit can be automatically generated for post delivery changes.
- The tool achieves the high quality, cost-effectiveness, and the efficiency.

**Disclosure**
Authors of this presentation have the following to disclose concerning possible financial or personal relationships with commercial entities that may have a direct or indirect interest in the subject matter of this presentation:

Min Chen, Xiangchen (Bob) Cui: Employees – Vertex Pharmaceuticals

*Acknowledgement*
 Appreciation goes to Tathabbai for his review and comments.
 Special thanks to Stacy for her supporting.