XML is a structured text format. To use XML in technical communication, you need an XML processor for functions such as editing, transforming XML into deliverables, and managing XML in a database.

An XML processor allows you to work with XML documents and their content. Most XML processors are automated tools, part of larger Enterprise Resource Planning (ERP) systems. They perform actions such as transforming content, updating databases, executing work processes, and delivering content to users. Some XML processors are aimed at specific XML actions, such as creating XML-based interfaces between applications. Technical communication requires XML processors that support the following functionality:

* Editing and maintaining XML documents.
* Transforming XML documents into usable formats, such as websites and printed documents. Transforming these documents may require tools that interpret one or more of these languages:
  + **XSLT:** a language for transforming XML documents into other text-based documents, such as another XML format for further processing or HTML for web publishing.
  + **XSL-FO:** an intermediate XML language designed to be transformed into PDF documents. Typically, an XSLT transform converts an XML document into XSL-FO, and a specialized application converts the XSL-FO into PDF.
  + **XPath:** a query language that lets you retrieve content from an XML document.
* Validating the structure of an XML document, guaranteeing its content is complete and contains all required sections. Languages that support validation include: XSD (XML Schema Definition), RNG (RelaxNG), and DTD (Document Type Definition). These languages define the structure of XML documents, including:
  + Tags allowed or required, including their ordering.
  + Attributes allowed or required.
  + Types of content allowed in each tag, such as text, other tags or other content.

For technical communicators, authoring is the most important aspect of using XML. A good XML processor allows technical communicators to focus on content quality while hiding the complexity of XML.