This piece originally appeared as the first in an ongoing series on Spend Matters Plus.

While contract lifecycle management, or CLM, is probably one of the most uninspiring acronyms in the procurement technology space, it is also one of the most important. That’s because proper CLM not only overlaps with both the sourcing (or source-to-contract, or S2C) and procurement (or procure-to-pay, or P2P) cycles for the strategic sourcing and transactional execution of “buy-side” categories, but also provides a partial foundation for other supply management processes.

CLM also influences the full source-to-settle (S2S)/source-to-pay (S2P) cycle. And, finally, it provides a partial foundation for risk management, performance management, change management and supplier management (as well as in-depth post-mortem reviews that increase organizational knowledge and effectiveness for years to come). Beyond procurement, enterprise contract management (also known as e-CLM) encompasses all contract types – buy-side, sell-side, partnership agreements, licensing, etc.). As an example, e-CLM can cover the transit of goods from supplier to customer or, as network modelers like to say, from source to sink.

However, this ongoing series will primarily focus on procurement-centric CLM as the focus of Spend Matters is on purchasing and supply chain. However, where relevant, it will indicate additional sell-side benefits that are offered by a CLM platform and the key capabilities it provides an organization.

Defining CLM

Wikipedia describes CLM as “the process of systematically and efficiently managing contract creation, execution and analysis for maximizing operational and financial performance and minimizing risk.” Others define it as an all-encompassing and all-inclusive process that starts with the identification of a business need and ends with a post-mortem analysis of contract execution.

Under this more extensive definition, it’s a 3-phase process that starts:

- "Upstream" (pre-contract) with need identification and category planning
- Continues with a (strategic) sourcing project that involves tender, negotiation and award
- And, ends with "downstream" contract execution processes that mostly live outside the contracting function (i.e., downstream in P2P, supply chain execution, internal audit, etc.) that focuses on change, performance and risk management

For example, the following depiction from CIPS is a good example of such an upstream pre-contract set of processes, a core set of negotiation/award processes and then a downstream post-contract monitoring and management processes. One of the challenges in such depictions is knowing where the processes directly involve the contract artifact itself or where they refer to related processes that integrate to the contract. For example, the terms "contracting" and "sourcing" are often used...
interchangeably, but they are really tightly intertwined processes that involve RFx documents (and related templates and cost element libraries) for sourcing and contract documents (and related contract templates and clause libraries) for contracting. These 2 sets of data follow the same design approach (i.e., rationalized data libraries that are designed for proper use/reuse into templates that are themselves reusable or directly into the end document), but also interact with each other.

As we continue on with our series, we will introduce our own simple process taxonomy that similarly focuses on core contract management within the "contracting box," but also many of the critical downstream process consumers of contract data that sit within process execution workflows. And, we will discuss a higher level analytic processes focused on performance management, risk management, etc.

While this more extensive definition may not be the typical one used by contract management (CM) vendors, and while it may make sense to keep upstream sourcing separate and hand off to a contract management solution only when an intended award has been identified, we are going to use this full definition because a successful contract is the result of successful planning. (In other words, there is more to contract lifecycle management than just Contract Management. And even though we will focus on defining the requirements for a contract management solution, it is important to understand where it fits in supply management).

This planning should include identifying suppliers who can – and will – meet any terms and conditions associated with making an award (such as insurance requirements, delivery terms, confidentiality, etc.), which should be defined during discovery and RFx and making suppliers aware of the contracting process to follow. So even though there can be a separation of technology platforms, there should not be a disconnect between the processes.

Where Does CLM Start and End?
From the viewpoint of a sourcing category manager, the full contract lifecycle management process is category management on steroids. It’s the process required to identify, source, procure, manage and realize the full potential of a category – any category. From the viewpoint of a CFO, the full contract lifecycle management process is a way to identify, realize and record the savings an organization needs. However, from our process-centric viewpoint, it’s an extended S2S/S2P process that requires some very specific technology, process and services support to be truly successful that goes (well) beyond what a sourcing suite or procurement suite vendor will offer you.
The requirements for this support are very extensive. A full description could easily take a volume. In fact, books have been written about the subject (including *Enterprise Contract Management* by Anuj Saxena that was co-sponsored by the IACCM).

However, since we’re primarily concerned with the technological (and platform) requirements, as opposed to the process and service support, so that we can review the existing solutions for contract management on the market, their maturity and their appropriateness to your needs, we’re going to focus on those.

But before we can do that, we have to first define what a CM solution is in the context of CLM.

However, we find that in order to properly define what a CM solution is in our context, we first have to map the solution space by defining the relevant process areas (i.e., pre-contract, core contract management, and post-contract) and then map them to the relevant software functionality areas that are relevant for those processes.