Conceptual Meta-Models: An Example Correlating Anthony's Triangle to the OODA Loop, the Deming Cycle, and the SDLC

Theodore Larson
University of Oklahoma

Jeremy Bellah
University of Oklahoma

Fangyu Du
University of Utah

Models are highly adaptable constructs. It is helpful to consider them in a solitary fashion as well as related to each other. A conceptual meta-model is an adaptable point-of-us application. In practice, the most successful models are simple and easy to portray. This presents an example of combining Anthony's Triangle with, the mutex of Colonel Boyd's OODA loop, Deming's Deming Cycle, and the ubiquitous Software Development Life Cycle when applied to an Analytics project's planning phase to successfully identify scope and methodology for the target audience. The end result is a continued appeal to the community to develop a sound set of core models that can be used for brainstorming with a broad base of professional understanding.

Keywords: conceptual model, information systems, decision sciences

INTRODUCTION

Conceptual model development is one of the most underlying methods of communication within most business fields. The general notion is to simplify complex ideas in such a way that they can be discussed and considered while any decisions maintain applicability to the real world. The end result of this is a plethora of models having been developed, with only a very few of them still being actively taught and used. Models such as the SWOT approach have legitimacy but no provenance whereas models like the MBTI have provenance but no legitimacy. This then devolves into a smattering of tools that many will recognize, but not always have a large amount of experience using.

There are typically two methods used for conceptual model development in the business world. They scale variably like Porter's Generic Strategies. The most successful are the ones that either seek to model every aspect of the real world, while still segmenting it into easier to calculate pieces or the models that seek to simplify the conceptualizations into the smallest number of entities in a very Shibui style.

However, just like Porter's Generic Strategies was modified by a novel application that suggested there was a third value-based strategy that did not lie on the continuum between differentiation and cost-

leadership, it is feasible to suggest that there is a third method of conceptual model development. This third method could be seen as a hybrid approach, but we posit that it is a novel form, seeking to provide as much real-world specificity as possible while still maintaining the abstraction benefits found in a simple model.

So, within this paper, we would like to suggest that the utility of using a conceptual model to simplify complex tasks without losing applicability to real-world decision making can be magnified by combining several models together to provide depth or interpretation, reinforcing a larger gestalt of each individual model while still utilizing a common set of simple, well-proven conceptual models.

LITERATURE REVIEW

There are many general business conceptual models that can be applied here. One thing that will be necessary for this particular approach to work will be to utilize models that are simplistic. For example, if a model to be functional requires a pre-printed worksheet, or drawing dozens of diagrams, it will be more difficult to conceptualize how the different parts fit in with other models. Rather, 'back of the napkin' type models that only require a handful of lines in order to be interpretable are more ideal. Similarly, to appeal to the widest range of people, common ones should be selected. As a result, the following three have been chosen, each reinforcing the other as a baseline model from which others can spring.

Anthony's Triangle has a long history of utility. It is a foundational model within the information systems sphere, as it deals with process design and automation. Anthony's Triangle was originally described as a pre-computer revolution method of design for factory organization (Anthony, 1965). However, as is often the case, the model was simplified and given its more iconic pyramidal shape and nominal appellation by Gorry, et al (1971), a few years later. It could be argued (and is the position of this paper) that it is widespread enough and genericized to the point that it is part of the common pool of knowledge for most managers. Full disclosure, the authors have already written about the shortcomings of this approach, but changing the structure of the model muddies the water compared to the overwhelmingly more popular pyramidal structure (Larson & Friesen, 2019).

Anthony's Triangle therefore will form a framework for the incorporation of three other models. The selection of these models was used as they have the same topology and intend, while broadly targeting different decision-making paradigms.

Boyd's OODA Loop

Colonel (USAF) John Boyd's seminal work in the arena of conflict revolved around the question of achieving superiority in air combat between two fighter pilots (Boyd, 2018). His model for decision-making is still taught as military doctrine in a number of applications within the military education system (Angerman, 2004). The central theme of the word is to not just make decisions, but that in conflict it's important to identify the opponent's decision-making process and interrupt it.

The general structure of the OODA Loop is that all decisions are ongoing processes divided into four phases as follows:

- Observation – The decision maker receives input triggering the need for a decision, either as an origin or a response to the outcome of the previous loop.
- Orient – The decision maker orients themselves within their environment as an actor that needs to respond to the observed trigger.
- Decision – The decision maker decides on the correct response to obtain an incremental improvement in their situation towards a broader goal.
- Action – The decision maker executes their decision.

As the original intention of the OODA loop was to model the split-second decision making of fighter pilots, it is not intended to be a proactive model for how to intentionally make a decision. Rather, it is a description of how situations change as a response to individuals' actions that respond to external stimuli. The utility of the model is not in making better decisions at the time of decision. The utility of the model is in preparation for understanding how to engage in pre-decision training and simulation to maximize the

instantaneous response for later moments of conflict. In more applied terms, a fighter pilot is not expected to combat an opponent by observing them, orienting, deciding, and acting; they are expected to prepare ahead of time by considering how they should react in hypothetical situations or to evaluate their previous performance during training exercises. This, then, is the utility of the OODA loop as it applies to the OODA Matrix: preparation and planning for implementation of conflict decision making.

Boyd's further prescription on the use of the OODA loop is to 'get inside' the opponents OODA loop. He suggests several methods of doing so:

- Being faster at decision making than the opponent, typically through training and increasing the repertoire of known situations that allow for immediate, optimal decision making,
- Interrupting the opponents OODA loop by changing the nature of the conflict,
- "Resetting" the opponent's OODA loop, by acting in a way that forces any of their phases to be invalid, forcing them to begin again at the Observation phase.

Deming's PDCA Cycle

W. Edward Deming developed the PDCA cycle in an effort to promote the idea of continuous improvement (1986). The goal was to try to universalize the idea that process improvement should be an infinite continuum from the past to the future, and that each iteration should be evaluated for improvement in the next cycle. The second goal was an early attempt at developing a manner of empiricism for the process of process design. Rather than a haphazard approach linked to any given managers prior experience and gut instinct, Deming was attempting to analogize the scientific method, but for business process design. What's interesting about the Deming Cycle is the fact that it is an epicycle, with discrete steps to manage each of the phases. While this complex nature would bely the process, if we abstract away the substeps in our metamodel, we're left with a simple 4-stage cycle.

The general structure of the Deming Cycle is:

- Plan – The Plan (emphasis added) is of primary importance in the Deming Cycle. By developing and using an initial explicit document, the rest of the steps should be easier to measure. It has five sub-steps.
    - Identify the Problem
    - Collect Data*
    - Set Goals*
    - Develop The Plan*
    - Obtain Support*
- Do – The actual action of the Deming Cycle is of lower importance. This will become important later, as it demonstrates where the Deming Cycle will fit into the larger meta-model. And even within this doing phase, the 5 sub-steps are largely only tangentially related to actual implementation.
    - Communicate*
    - Execute
    - Collect Data*
    - Document*
    - Correct*
- Check – The check phase is, itself, an interesting microcosm of business analytics and predicts many of the analytics frameworks in current use. The phase primarily orients itself around what are now standard analytics practices, however an important point is to note that these are intended to be process-specific rather than enterprise-wide.
    - Analyze*
    - Identify Patterns*
    - Evaluate*
    - Identify Improvement*
    - Standardize*

- Act – Finally, one of the things that sets the Deming Cycle apart from similar models is the final phase. Rather than just imply that there will be a cycle and the improvement there in, the Deming cycle specifies that the Act phase is about adjusting the entire cycle to improve on the efficiency and effectiveness of the end result.
    - Develop Improvement Plan*
    - Implement Plan*
    - Monitor*
    - Document*
    - Standardize*

In one sense, the applied nature of process design has made the Deming Cycle seem a little more antiquated in the phase of newer design cycles like the SDLC or the BPML. However, one important thing to note is that the newer models tend to be more simplistic and emphasize direct action, the Deming Cycle has a vast majority of its steps being observational/managerial steps rather than direct action, as indicated by the asterisks. This is the reason to list out all of the sub steps in this early explanation, to illustrate that twenty-three of the twenty-five are tangential to the actual activity.

The Software Development Life Cycle

The SDLC is a model with low provenance. While it is known to have come into fruition some time between the presentation of the Waterfall model (Royce, 1970) and the late 1980s. It is possible that the SDLC came into being as a memory error between people confusing the Waterfall model with the Deming Cycle. Regardless, the SDLC is intended to provide an overview of different phases of the software design process and quickly was adopted as the Systems Development Life Cycle with the added confusion of the same acronym. It is the authors opinion that several facts about the SDLC are true: it is no longer possible to trace the exact route and timing of its development, its lack of a referential trail places it squarely in the public domain, it should be common knowledge within the Information Systems and Business Analytics community, and because the SDLC is of vague origin then any reasonable representation is just as viable as any other reasonable representation. SDLC will be used interchangeably between Software and Systems, and will be applied to any analytics project as either.

The general structure of the SDLC is:

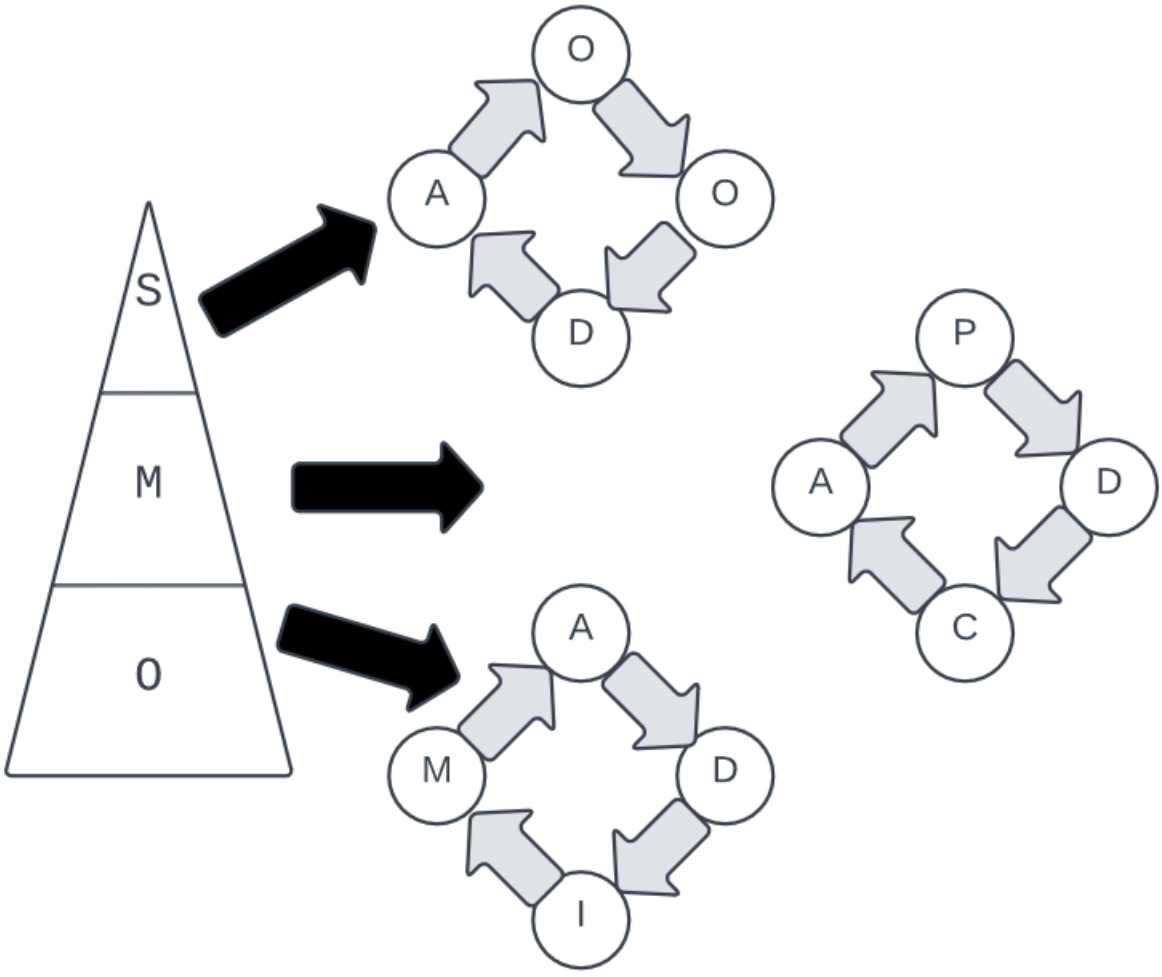
1. Analysis – The analysis phase can be structured as a process with an input of the current situation with regard to a dataset, an organization, or a problem. The output is the formalized, structured description of the problem as a declaration of its primary import toward the current situation.
2. Design – The design phase can also be structured as a process with an input of the previous formal decision structure and the output as a design document that delineates the intended solution process.
3. Implementation – The implementation phase has an input of the design document and an output of the execution of the design document. However, this phase has the most nuances, most commonly having the prototyping pre-step separately listed or the testing post-step separately listed. They will be combined together for this analysis to stay topologically similar to the other models. But a one-to-many mapping between the other models and this one would be feasible.
4. Maintenance – The maintenance phase of the SDLC has the new execution as the input and, over the course of time, eventually has the output of a new situation that needs to be analyzed for improvement.

Again, this cycle is difficult to establish literary provenance for, however it has widespread usage within the common knowledge of programming, systems, and academic professionals. The goal is to discretize the specific steps of any given task in order to make executive or role distribution easier to handle.

Anthony's Triangle makes up the primary model that will drive this meta-model. However, each of the cyclical models presented will form a set of mutually exclusive secondary models that are determined by either the role of the target or the developers in line with the use of Anthony's Triangle.

Framework

FIGURE 1
CONCEPTUAL META-MODEL – ANTHONY'S TRIANGLE TO OODA, PDCA, AND SDLC BY ROLE



Anthony's Triangle is intended to be used to discretize the decisions that may be encountered by the different levels of role within an organization. However, this decision-making factor can be looked at in two ways when it comes to the development of any analytics project, with the assumption that any analytics project will always result in the publication of results in some capacity to aid in decision making. That is, analytics itself is a field about making decisions. So, when Anthony's Triangle is applied to Analytics, the question becomes: do the organizational levels apply to the producer of the analytics output (the analysts) or the consumers of the analytics output (the viewers)?

A tautological answer is that, since conceptual models are abstractions of the real world, the intent and application are mutable. In essence, a model is merely a medium for communication, and therefore the presumption of entities within that medium is made by the communicators. Therefore, it can easily be said that the model can apply to either the producers or the consumers.

When considered with respect to the consumers, the model looks at the decision-making tasks of the people who will be using the final result:

- Operational – Operational viewers will be primarily focused on lower-term solutions. The predictions or the visualizations should produce direct and immediate answers to immediate

problems, likely utilizing a large amount of quantitative data to converge onto single, immediate decision points.

- Managerial – Managerial viewers will be primarily focused on more nuanced decision-making scenarios where outcomes are going to be more ordinally focused or otherwise developed with some level of agent interpretation involved. This means that rather than directing to a specific, individual answer, a range of answers or limited what-if analysis needs to be accounted for.
- Strategic – Strategic viewers should be primarily focused on decisions with no clear answers, that is the heuristic between problem and solution has no predictive guarantee. This is where satisficement might enter, or where large sections of cascading decision trees might be the optimal outcome. Ultimately, no answers will be provided, instead a myriad variation of methods to analyze data to come to unstructured solutions.

This approach is an excellent method for looking at any given analytics solution and allowing the developers to determine appropriate outcomes. The development team can determine the nature of the questions being asked of the analytics solution and the role of the questioners in an effort to target their solutions more narrowly. For example, an entry level employee with only a few months of experience should not need to spend more time learning a complex analytics solution than they spend solving a limited set of narrow solutions. A better outcome would be several targeted tools that provide direct and unmitigated answers to structured problems.

When considered with respect to the producers, the model looks at the roles within the development team:

- Operational – Operational level team members would be the ones who are responsible for the actual development of the analytics solution. Regardless of whether they're part of the modeling, presentation, or tools roles (Larson & Friesen, 2019), in this context they are concerned with the immediate tasks of developing a directed product.
- Managerial – Managerial level team members, in this context, are the team supervisors who are responsible for longer range planning. They are responsible for determining the functioning of the project on a longer term basis, considering the impact of the project as a whole, and directing the operationalization of the project goals over a span of weeks or months.
- Strategic – Strategic level team members, in this context, are the stakeholders outside of the team (but likely within the organization) who are responsible for making broader decisions like 'what questions need to be answered?' and 'what organizational goals can be enhanced with decision-making analytics solutions?' This does not make them part of the process of the development of the team, rather it makes them an intricate part as they give broader context to the more narrowly defined managerial tasks.

Teaching analytics as a discipline, or mentoring new analytics professionals provides an excellent place to utilize this application of Anthony's Triangle to make sure that everyone understands what their role is in applying analytics technologies to any given solution. It is a different scope than other applications, but is equally useful in understanding the approach to robust analytics development.

While both the producer and the consumer are valid frameworks for this meta-model, it is more cleanly demonstrated with the producer side. For example, when developing a design document questioning what roles each level of stakeholder will be making in the development of the task, what process should be made in doing so?

Combining Anthony's triangle with the three specified cycles leads to a novel solution that distinguishes roles, provides unique terminology for allowing that decision, and allows for either strict or freeform analysis as it's applied.

For the sake of this example:

- Strategic level decision makers will use Boyd's OODA loop.
- Managerial level decision makers will use Deming's PDCA cycle.
- Operational level decision makers will use the SDLC.

Application of the OODA Loop to Strategic Development

This layer does not require, but is more justifiable when a base assumption is made. That is, organizational conflict. Within the context of the open market, competing organizations are always in conflict. And this conflict can be modelled similar to physical combat, a la Boyd's OODA loop. While the original context is to make immediate decisions to 'get inside' the opponents OODA loop, thereby disrupting it, there are immediate parallels. A modern sensibility is to 'disrupt' a market, which is the concept of simultaneously 'getting inside' all competitors OODA loop's with a completely unforeseen move. A classic example of this is Apple's release of the original iPhone (Watts & Augiere, 2022).

So, presuming conflict is one consideration of inter-organizational interaction, it's feasible to look at the available tools to determine what is available. In the case where the tool is the broad use of analytics, a simple matrix suffices:

- Org A has no or outdated Analytics, Org B has no or outdated Analytics: Competition occurs with historically standard norms.
- Org A has modern Analytics, Org B has no or outdated Analytics: Org A will have a competitive advantage, and Org B's strategy will be easier to decipher for Org A than vice versa.
- Org A has no or outdated Analytics, Or B has modern analytics: Org B will have a competitive advantage, and Org A's strategy will be easier to decipher for Org B than vice versa.
- Org A has modern Analytics, Org B has modern Analytics: Competition will occur on an even basis, and there is a reasonable expectation that either org can decipher the other's strategy, leading to an arms race in the development of newer analytics tools.

Thus, it should be clear that the direction of an overall analytics strategy is a strategic goal that has direct impact on the relative power dynamics between any two competitors (and certainly extensible to any number of competitors). With this perspective, it is easy to see that Boyd's OODA loop could play a pivotal role in the direction of longer term analytics strategy.

- Observe – Determine the relative decision making capacity of competitors analytics
- Orient – Place the relative decision making power of analytics frameworks within the current technological ecosystem, both what is extant as well as what is feasible.
- Decide – Make a determination of organizational analytics strategy.
- Act – Direct the development of this strategy.

The counter to this good outcome would be the equivalent negative outcome. This would be where during any of the stages, a fundamental shift in the opponents capabilities changes without warning. For example, Reid reports that subsequent to the success of OpenAI's ChatGPT, many startups attempted to market their own products but were immediately priced out of competition by OpenAI's ability to offer their product for an exponentially lower price (although, it begs the question whether this is a technical competitive advantage or simply a first-mover advantage) (2023).

Application of the PDCA Cycle to Managerial Development

It has already been noted that the Deming Cycle primarily focuses on managerial tasks. Rather than determining direction at the strategic level or actually performing the implementation at the operational level, the Deming Cycle is purely a managerial level tool. As a result, each of the phases of the Deming Cycle should revolve around the improvement of the previous cycle, or benchmarking for future cycles.

- Plan – The planning phase may be analogous to the SDLC's analysis and design phases, however it is more strictly targeting the framework for analysis of the development of the analytics tool, rather than the direct analysis and design activities. Regardless, some crossover would be reasonably expected, as the managerial level and the operational level will certainly interact.
- Do – Again, the overlap with the SDLC's Implementation phase or the OODA Loop's Act phase is certainly obvious. However, in this context, the entire PDCA cycle could be said to be operating within the Act phase of the OODA Loop, and the overlap is similarly appropriate.

- Check – While the operational level team members would be focused on the maintenance of the actual solution and insuring database connectivity, publication availability, bug-fixing, etc., the managerial level team member should be more focused on process analysis and improvement of the entire process for the next cycle.
- Act – Aside from possible confusion about similar names in different sections, again it should be noted that the Deming Cycle culminates in the implementation of the process oversight. This act phase is actually in preparation for the Analysis phase of the next SDLC cycle.

Not enough discussion has been had about the application of the various business cycles to Analytics as an industry, the various analytics tools, and others. Such discussion certainly exists within the literature, but it is nowhere near as robust as the yearly output of generalized management concepts. Care should be taken to engage in self-reflection within the analytics field, as the field continues to mature.

Application of the SDLC to Operational Development

It should, at this point, be fairly evident that the SDLC is a practical, implementation-oriented tool. So, at this point, it's important to remember that the use of conceptual modelling is merely an abstraction of reality. It is not an actual distinction, but rather a set of conceptual tools for reducing the complexity of any task and allowing individuals to recognize specific patterns or roles within that task. Even still, the steps can be laid out:

- Analysis – Within this meta-model, the analysis phase is entirely focused on the operationalization of a given problem. It is primarily focused on tool selection and problem framing, and less so on broader issues like needs analyses.
- Design – Within this meta-model, the design phase resembles a more traditional SDLC utilization, with the development of a design document, benchmarks, etc.
- Implementation – Again, within this meta-model, the specific prototyping or testing apparati are possibly determined by different roles. However, the actual coding, tool use, and generalized production of the end result are handled here.
- Maintenance – This is, possibly, the largest benefit of the process. Estimates vary, but they can go as high as 90% in terms of the relative time and/or cost spent in the maintenance phase. Targeting the actual maintenance to the operational level team members systematically frees up managerial and strategic stakeholders to perform their intended roles more effectively.

CONCLUSION

The utility of this meta-model is variable. One approach to product development would suggest that there can never be too much planning. However, just like any categorical statement, there comes a point where that position becomes farcical. So, the question becomes when this model hits that point.

So, there are two suggested modes for implementation. On the more crystallized framework, organizational processes can be designed and documented that specifically lay out the roles and responsibilities of each of the levels at each stage of their respective cycles. The disambiguation between models allows the longer-term planners, non-technical stakeholders, and unstructured decision makers to turn the entirety of the actual development into a black box. The disambiguation also allows the managerial level members to specifically ignore questions of implementation and directly target oversight and performance management. And all of this should still leave the operational level employees to direct, structured tasks within the commonly accepted SDLC framework.

The more fluid framework would allow any team member to describe where they are with a quick, 'back-of-the-napkin' sketch. Even within other design paradigms, being conversant in these standard management models would make communicating the position and current goal of any tasks much simpler.

Of further research interest would be the variability of time scales. Since these are all cyclical applications, the question becomes one of synchronization. The classic application of Anthony's Triangle says that Strategic level scope is in the multiples of years, Managerial level scope is in months to a year,

and the operational level scope is minutes to days. So, to what extent do these models synchronize in practice? Does the OODA Loop cycle predominate the rest of them? For example, should a competing organization 'get inside' the local organization's OODA Loop, does the reset back to Observe precipitate a reset across the others? The decision was made to avoid postulating about it, as empirical, observation of real-world scenarios where this meta-model (or a similar one) is used would be more effective than guesswork.

Once again, the goal of this paper is to demonstrate the utility in the development of a common grammar of models so that professionals can more easily communicate with each other in a manger that provides the greatest balance of flexibility and documentation. The utility of the discretization provided by most conceptual models makes this feasible, and simple models can become as complex as needed through layered application.

In short, this paper has demonstrated the feasibility of combining, with varying levels of abstraction, the utility of specifically combining Anthony's Triangle with Boyd's OODA loop, Deming's PDCA Cycle, and a generic SDLC.

REFERENCES

Anthony, R. (1965). *Planning and Control Systems: A Framework for Analysis*. Boston, MA: Harvard University Press

Boyd, J. (2018, Reprint). *A Discourse on Winning and Losing*. Air University Press.

Deming, W. (1986). *Out of the Crisis*. MIT Press.

Gorry, G., & Morton, M. (1971). A Framework for Management Information Systems. *Sloan Management Review*.

Larson, T., & Friesen, D. (2019a). *Information Systems for Non-Technical Managers*. Cognella.

Larson, T., & Friesen, D. (2019b). The Anthony Triangle and an Analytics Framework: Developing a Business Analytics Curriculum Conceptual Model. *Journal of Management Science and Business Intelligence.*

Reid, R. (2023). *Has the generative AI pricing collapse already started?* Retrieved from https://arstechnica.com/gadgets/2023/03/has-the-generative-ai-pricing-collapse-already-started/.

Royce, W.W. (1970). Managing the development of large software systems. *Proceedings of IEEE Wescon*, pp. 1–9.

Watts, B., & Augier, M. (2022). John Boyd on competition and conflict. *Comparative Strategy*, *41*(3).