



Why Estimates Never Work

By: Dr. Colin E. Beech

Introduction

Estimates are one of the cornerstones of contemporary project management. Junior Project Managers and Clients alike often stare in wonder at a fully developed work breakdown structure that relies on estimates, trusting almost absolutely in its neat rows and columns of deconstructed work packages. Its secrecy seemingly guarded, it is often some years as a practitioner before a PM is trusted with creating estimates of their own within their organizations, and even then, almost always in strict accordance with the principles outlined by the Project Management Institute, or PMI. Most often associated with the “Waterfall”, or Big Design Up Front approach, arriving at an estimate is one of the most crucial elements to the success of a project since it comprises the very essence of cost control and measuring the performance of the organization, a hallmark of modern management methodologies. Why then, for all the fanfare, are estimates rarely effective at accurately capturing costs and time to completion? This article will examine many of the uncommonly known pitfalls to estimation as a best practice, avoiding reliance on them altogether for alternative guardrails and safeguards for the performing organization instead.

Classical Methodologies

Why do estimates often fail to be accurate? A quick review is in order to understand the underlying structures of estimates and estimation. The PMI suggests that there are several major methods of estimation in the classic “Waterfall” life cycle. One of them is the “Historical” approach, where the actual hours spent on projects of previous size and scope are reviewed, and a similar timeframe is offered as an estimate. A “Bottoms-Up” estimate is one where each element of scope is analyzed at a granular level, estimated by a subject matter expert, and these elements all roll up to one cohesive duration determination. “Parametric” estimation is possible when projects follow a life-cycle phase structure so consistently that a formula can be derived that allows for extrapolation of the entire life cycle from one known portion. For example, if engineers offered a bottoms-up estimation of creating a new sort of “widget” with a reasonable level of confidence, say, one hundred hours, and the initiation of such projects reliably takes about ten per cent of the engineering build time, with analysis & design taking twenty-five per cent, and so on down the line, a parametric estimate may be made. “Three-Point” Estimation, also known as PERT, and the final major PMI method discussed in this paper, sees the Project Manager making three estimates: a best-case scenario, a worst-case scenario, and a most-likely scenario, allowing the Project Manager to weight the average and take it as the estimate of record (*Project Management Institute, 1996*).

Let’s examine where these methods often go awry. Leading off, there’s the very nature of how these estimates are treated within their organization. The PMI is very careful to mention with all of the estimation techniques it presents in the PMBOK that estimates are offered in the spirit of *variance* – the timeline of the estimate is just that, an estimate, and different techniques will offer differing levels of variance, or error (*Project Management Institute, 1996*). Parametric estimation may come within twenty-five per cent variance (pretty good), and this may suffice for many organizations. A bottom-up



estimation method widely regarded as more accurate, may come within ten per cent of its projected end date. In all these cases, the problem with variance is that even if it is accurate, the variance may lay *beyond* the mean, or likely outcome, and many organizations will take this mean date of the estimate as the actual deadline. PMI would be the first to mention that estimates are not meant to be used in this fashion, and that the whole purpose of Earned Value Management, or EVM, is to track performance using performance indices, and use these status reporting criteria to suggest early on if a project is likely with acceptable variance or not, and even then, that it is the outside range of variance *plus* the contingency reserve that should be communicated as a “deadline” to management (*Project Management Institute, 1996*).

If that were all there was to it, then Project Managers across the land would simply emphasize these elements to their C-suites, who would recant, and follow the “rules”; that would be the end of it. As the title of this article implies, however, there is far more to it than that. As far back as 1979, when computer programming was beginning to be a discipline and industry ruled by norms and best practices, Douglas Hofstadter examined the overwhelming difficulty of creating reliable estimates. So widespread was this issue, that he penned the now infamous *Hofstadter’s Law*:

“Hofstadter’s Law: It always takes longer than you expect, even when you take into account Hofstadter’s Law” (*Hofstadter, 1993*).

Hofstadter wasn’t the only one to spot this trap. Fred Brooks noted in his *The Mythical Man Month*, now a seminal work in all things IT and estimation, that management as a group regularly makes errors regarding resources, their allocation, and expectations of productivity all the time. He notes the specific example of what happens when managers sense that their estimated timeline is at risk, so they “crash” the project (as PMI would recommend) with the addition of resources. Counterintuitively, this tends to make the project *even worse*, as additional resources need time for training and to come up to speed on the project in its *sui generis* context, and that this is often at the expense of the team that is already on the ground. Just as one cannot get a baby in four and a half months instead of nine by tasking two women to be pregnant at once, Brooks outlines, neither can teams make deadlines even when crashed in this way (*Brooks, 1974*).

Underneath these arguments is the problematic nature of “deadlines” themselves. In the kinds of organizations that the PMI takes for granted, adequate time is given for accuracy, risk management, team and stakeholder feedback, and other niceties that are rarely found in the majority of organizations conducting activities in a crowded marketplace. Sociologist Mary Douglas notes that these organizations, or institutions, are tasked with making “life or death decisions”; true, individuals are performing the rote calculus, but it is always in their capacity as a member of the larger organization and its pressing goals (*Douglas, 2012*). While the project manager may think of a deadline produced by their estimate as comfortable and bound by the triple constraint (*Project Management Institute, 1996*), the C-suite of their organization may be facing staffing cuts unless a deliverable may be invoiced by a certain cutoff date; in this sense, the C-suite hears the “deadline” they want to hear, the rosy estimate of the Project Manager. Because economic necessity drives much of the organizational decision-making in this way, the reality is that project managers face enormous pressure to make deadlines that serve these “life or death” needs of the organization in a way that overrules calculated, rational methods. Ask many seasoned project managers in a variety of industries, and they’ll tell you that more than once they’ve



had to reverse-engineer an estimate to conform to an external constraint deadline, directly counter to PMI best practices and the very notion of the Triple Constraint. This defeats the purpose of estimates and schedules based on them, relegating them to mere busywork and invoking the specter of the “Death March.”

Interpretive Flexibility and Rhetorical Closure

Industry anecdotes and domain-based publications aside, what are some of the driving forces behind this phenomenon of poorly performing estimates? A recently formed field in the social sciences and the history of technology, the Social Construction of Technology, or SCOT, framework can offer a significant amount of insight into why estimates are intrinsically problematic. Bijker and Pinch argue in their now seminal essay on the social history of the development of the bicycle that product engineering is a messy, socially-situated process that takes time to unfold. “Customers” and “Designers” are but two roles in a larger ecosystem of markets, laboratories, workshops, third-party reviewers and industry pundits, and so on; design activities take place in this broader milieu, and as a technology is created, it passes hands, as it were, from one group to another to be scrutinized. The very idea *a priori* that one could accurately determine how long this process would take is merely a philosopher’s folly from this point of view (Bijker et al., 2012).

Bijker & Pinch further define the development of technologies into two discrete phases: *Interpretive Flexibility* and *Rhetorical Closure*. Interpretive Flexibility is characterized by a period of various different interpretations and understandings of a new technology as it makes its rounds in the design and user communities and beyond. The technology itself may be broadly and variously viewed as anywhere from fundamentally flawed in its very premise, to literally the salvation of humanity, and all points in between. The electric vehicle has been experiencing this moment for almost ten years now, as an example that comes to mind, and like the bicycle that Bijker and Pinch describe, the definition of a “good” example of the technology varies tremendously from person to person, and group to group. The practiced project manager will understand here that stakeholder dynamics are being discussed.

Should a bicycle have two wheels, or would three be a better platform? What is “better” in this sense? More stable? Higher speeds? Should the wheels be the same size, or different? What should the frame be constructed of? What is the best material for creating a wheel and its contact point with the road? Various fascinating examples are all discussed by Bijker & Pinch where very different incarnations of the bicycle are created, refined, tested, applauded, rejected, and other various reactions. One of the better anecdotes is when a young Mr. Dunlop appears at a bicycle race in Britain in the early 20th century with wheels shrouded in pneumatic rubber of his own invention, to the enormous laughter of all participants who surely *knew* that proper wheels are made of wood and steel exclusively. Mr. Dunlop, a merely average athlete, won seven of his eight races, and went on to form a tire empire; this is the familiar story of the technology disruptor, and it emphasizes the other primary state of technological development that Bijker & Pinch identify: *rhetorical closure* (Bijker et al., 2012).

Rhetorical closure occurs when a particular breakthrough or development is so invaluable that it becomes a quintessential feature of the product going forward; to not include it would be considered irresponsible neglect on the part of the designer. “Good” bikes have rubber tires like “good” EV’s have a range beyond 300 miles; these are now “industry standards.” This is the very essence of rhetorical closure: the end-state of a design and requirements discussion that a good project manager strives



toward on the part of project stakeholders, where any further discussion is moot, and all parties are agreed and satisfied they've been heard. The positive side of this characteristic is stakeholder agreement, the predictability that comes with it, and of course (spoilers), good estimates. The negative aspect of rhetorical closure is that these elements are now all accepted as “fact”, and while it may once have been rooted in disruption of its own, it would take another disruption to dislodge any design elements from the model; they are now “standard issue” (see Figure 1). In this way, one learns that product development, adoption, and expectations are more political than they are rational or objective (*Winner, 2020*), a full-throated answer to why estimates that rely on objective methods can fall so woefully short: there's no accounting for politics.

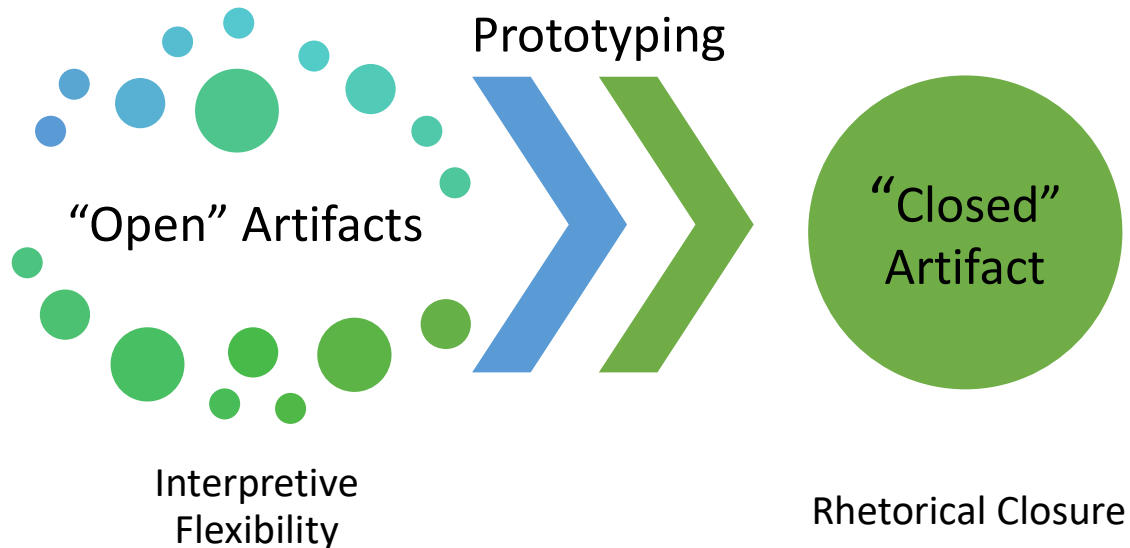


Figure 1: Artifact Closure via Iterative Prototyping

Achieving rhetorical closure requires giving conversations and user experiences time to play out, to gel, and become broadly taken up. This is the reason why Bijker & Pinch conclude their work to make the case for *prototyping*; one cannot avoid the path from the one state to the other, and design in a vacuum will only hinder engineering as a social process. Teams and departments need to work together through multiple iterations, and then welcome as well feedback from customers and other users. Rhetorical Closure cannot occur until all these interactions have played out; prototyping, as a practice, deliberately opens and accelerates these processes. In her history of Silicon Valley and the role geographical proximity played in its rise to prominence, Saxenian noted that corporations based on the East Coast performing software design were hierarchical, insular, and atomic, whereas West Coast entrepreneurs were deliberately collaborative with one another in a decentralized web of service offerings that built on one another in a novel form of collaborative capitalism (*Saxenian, 1996*). In this way, Silicon Valley created products with enhanced, prescient rhetorical closure that came to dominate the industry as a whole.



At the opposite end of the spectrum, the organization that does not prototype vigorously invites the possibility of creating a *reverse salient*. First identified by Thomas Hughes in his history of the advent of large-scale electrification grids, reverse salients occur when teams work independently of one another on major components of a large-scale system (*Hughes, 1993*). Take, for example, the construction of an elaborate web portal with many fields requiring validation, and a database backend. Each team of analysts, developers, and architects might focus on their own exclusive product development to the point that when it comes time to integrate the two elements, the project manager faces the “showstopper” risk that the portal uses a JSON format messaging layer, while the backend team was expecting data via XML. This seems like a laughable bungle, both sides of the bridge not meeting over the river like a *Far Side* cartoon, but such reverse salients can and do occur when the external pressures of the organization drive teams into a mode of hyper-focus without mandatory prototyping.

The Iterative Design Cycle

SCOT gets right to the heart of the matter as to why “scope creep” happens so frequently, and necessarily, on major technology projects. As a technology makes the rounds, as it were, different user groups respond differently to the product, its context, its overall number and content of use cases, its means of production, and a variety of other factors. The time it takes to move completely from a state of flexibility to one of closure is more a function of the number of these groups and their divergent goals than it is rote engineering, though that is surely a factor. So much so, speaking of engineering, those groups will similarly often add feedback resulting in scope changes as major infrastructure elements are considered holistically.

This “scope creep”, which has a negative connotation according to the PMI and widely throughout the industry, may be thought of in SCOT terms as a lack of stakeholder management and feedback, and of the failure of “Big Bang” style planning in general at capturing broadly divergent viewpoints (*Project Management Institute, 1996*). Rethinking scope creep mitigation instead as deliberate communication between areas of intersection and alternative viewpoints of a problem would seem to be the order of the day. Critique, argue Connor & Irizarry, in a rapid prototyping setting offers the methodological tools to not only hold open conversations about design, but to catalyze them with deliberate, targeted communication relevant to design with rhetorical closure as its explicit goal. They are careful to define critique in a way that it is not confused with rhetoric as feedback; critique is done from a stance of equilateral power with the specific intention of informing the designer of previously unconsidered aspects of their work, whereas feedback is often done from an asymmetrical position of power where the designer is compelled to change their design in a manner contrary to their intentions (*Connor & Irizarry, 2015*). The irony of this exercising of power as feedback is that is usually coming from the benevolent goal of cost control, but in the end, stymies the urgent process of discovery that leads to products that pass the test of rhetorical closure; it is these products, tested and refined, that gain desperately needed market share.

The New Cost Control: In Defense of “Hello, World”

Okay, you may be saying, estimates are bad at accounting for the social aspects of design, and iterative prototyping is better at generating true critique in regular intervals; businesses must still be managed!



How can controls be implemented to govern a social design process, especially when costs are on the line? In a phrase: decouple cost controls from estimation, and simply stop estimating. Let me explain.

Kaizen has rocked the operations world with its practical, well-thought-out methodology that stresses an open and inclusive culture of innovation, with no value addition being too small. Small is beautiful, in the Total Quality Improvement model, and it creates an ease of overall implementation. It can be communicated fairly easily, and creates a single node of comparison across a larger whole. Its effects will be modest if it goes awry (barring bad luck), but its value will be palpable if it is indeed effective in a measurable way. Kaizen methods turn over every stone in the search for value in a process, end-to-end, in a way that Bijker & Pinch would surely approve of in the quest for product closure.

These principles have moved from an operations arena into more project-based thinking with the advent of agile software development and lean design principles. There is an important gap which must be jumped first: project management is very different from operations management for reasons well explained by the PMI itself, chief among them that projects have a beginning and an ending, whereas operations go on in perpetuity (*Project Management Institute, 1996*). If one considers initiation, analysis, & design to be habitual activities with repeating structures, however, it makes sense that Kaizen principles of increasing value in process chains while shedding waste might apply. This is important, because otherwise this is an “apples to oranges” case for optimization. Having made the intellectual leap however, one sees that managing incremental changes is vastly easier than “big bangs” or “rapids” (a series of small waterfalls) even in project management culture, and in particular, in driving designs to to a state of rhetorical closure.

Contemporary authors and leaders in the industry like Philip Holt take this idea even further. Holt contends that Lean is a state of being for the performing organization. In this way, the dichotomy between “operations” and “projects” becomes rather blurred: any existing value-chain or operation within an organization can benefit from being leaned, and projects become less gargantuan simply by virtue of having an organization that is in a state of constant incremental improvement. Essentially, nothing goes awry to the extent that a “project” becomes necessary if lean becomes a way of life (*Holt, 2020*). It’s a charming philosophy, but the reality is that new systems need to get made all the time in virtually every organization; how to apply lean in this way, and with cost controls in mind?

The conventional, Big-Design-Up-Front approach would have the organization select a Champion, draft a charter, and select a Project Manager, who would then in turn secure a few resources and SMEs who would go about making an estimate. An agile, lean project start-up would begin instead with a prototype to satisfy a basic need expressed by any key stakeholder who has decision-making authority within the organization to commit resources to its construction. Ideally, with roughly the same amount of expenditure in terms of resources, a prototype creates a tangible, functional product that would have otherwise simply been an ethereal estimate. Prototypes “work” when they generate visceral, tactile comprehension with few words spoken in the service of understanding, but a great many in terms of *possibility*; this is the essence of moving towards rhetorical closure and stable product release.

Many programmers intuitively understand this, as their careers often have begun career with the simple “Hello, World” exercise, the initial Use Case that stresses a stack end to end in service of a brief, friendly message. Prototypes are simply seeding this initial sense of wonder from the get-go by understanding the value-proposition of the prototype itself, and then making the rounds with first internal, then



ultimately external stakeholders in a constant cycle of iterative product development (see Figure 2). Project Managers exercising cost control have no choice but to let this process play out, but can allow it to do so within the major guardrails of quarterly release schedules, drop-dead dates informed by the marketplace or other arenas (elections, perhaps), and the operating capital available to the company.

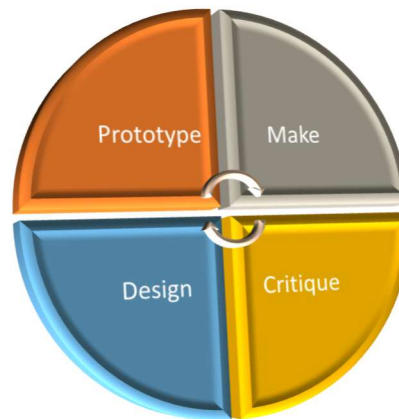


Figure 2: The Lean Prototype Cycle

Stated another way, a contemporary, lean organization that didn't have any existing operations or product offerings would almost certainly be starting in an incubator-like setting, or perhaps literally. A business plan was formed that stated the value proposition of a certain product idea or prototype, enough so that some sort of initial investment was made. The challenge now faced, with a prototype in hand, is to perform *make* activities that enhance existing aspects of the prototype; this is where the engineers earn their money as they expand on the value of a product with additional Use Cases, adding features, "knocking off the edges", or all of the above, with management watching from a distance with dwindling investment capital in mind. Critique, as Connor and Irizarry have described, is used by management to give their own analysis of the value proposition of a product, always driving towards a Minimum Viable Product they feel meets release criteria to bring in additional capital, either through investment or sales (Connot & Irizarry, 2015). The critique process fuels additional Design ideas, moving down all the newly found lanes of interpretive flexibility (with the Project Manager brokering dialogue between stakeholders and designers), until enough of the conversation crystallizes with the creation of a new prototype, and the whole process begins again.

Eventually, with costs carefully measured against a clear course towards an MVP candidate, the "widget" will crystallize into a value proposition to those beyond just the organization's walls. An *operation* emerges as a stable product offering and/or value-chain, and it is here the organization has the happy problem of continuing to revise the "widget" with lean principles in mind as Holt describes, or to fund activities on more new "widgets" to be managed in the same way (Holt, 2019). What of complex, major projects, like a federal contract to build an Aircraft Carrier? Candidates who bid on such contracts are usually already in the Aircraft Carrier business, and while they may use lean cycles to mature needed elements to be competitive, they usually already have a value proposition in hand. In this sense, organizations rarely move from true prototype to Enterprise-level products, instead arriving at maturity via iterations, even if they are the sporadic lurches of a poorly planned "Waterfall" project.



Conclusion

In this article, many of the challenges that estimation methods have historically faced were reviewed resulting in their poor accuracy. Some of the social underpinnings of this phenomenon were discussed, particularly the role design plays in driving an artifact from interpretive flexibility to rhetorical closure, borrowing from the SCOT framework. Ideas from Kaizen, or continuous quality improvement were discussed, and how many of these principles have been incorporated into project management as well. Finally, with the lines between “projects” and “operations” blurred, the role of estimation drops away and is replaced by the challenges of prototyping towards an MVP in the contemporary organization, and exercising lean, agile methodologies from there to further drive value propositions and ROI on the part of the performing organization.

Works Cited

Bijker, Wiebe E., et al., editors. *The Social Construction of Technological Systems: New Directions in the Sociology and History of Technology*. MIT Press, 2012.

Brooks, Frederick P., Jr. *The Mythical Man Month and Other Essays on Software Engineering*. Addison Wesley Longman Publishing, 1974.

Connor, Adam, and Aaron Irizarry. *Discussing Design*. O'Reilly Media, 2015.

Douglas, Mary. *How Institutions Think*. Routledge, 2012.

Eggers, William D., and John O’Leary. *If We Can Put a Man on the Moon: Getting Big Things Done in Government*. Harvard Business Review Press, 2009.

Hofstadter, Douglas R. *Godel, Escher, Bach: An Eternal Golden Braid*. Vintage Books, 1993.

Holt, Philip. *The Simplicity of Lean: Defeating Complexity, Delivering Excellence*. Management Impact Publishing, 2019.

Hughes, Thomas Parker. *Networks of Power: Electrification in Western Society, 1880-1930*. Johns Hopkins University Press, 1993.

Project Management Institute. *A Project Management Body of Knowledge*. Project Management Institute Communications Office, 1996.

Saxenian, Annalee. *Regional Advantage: Culture and Competition in Silicon Valley and Route 128*. Harvard University Press, 1996.

Winner, Langdon. “Do Artifacts Have Politics?” *Emerging Technologies: Ethics, Law and Governance*, Routledge, 2020, pp. 15–30.