Project Portfolio Management with "Measure and Match" and the "IC-Map"
A Portfolio of Illustrations

John Helm

This document contains the key figures that summarize two tools I've developed to help with project portfolio management. Several of these diagrams are not very self-explanatory and therefore require some supporting context. My purpose here is to redress this omission by providing an annotated portfolio.

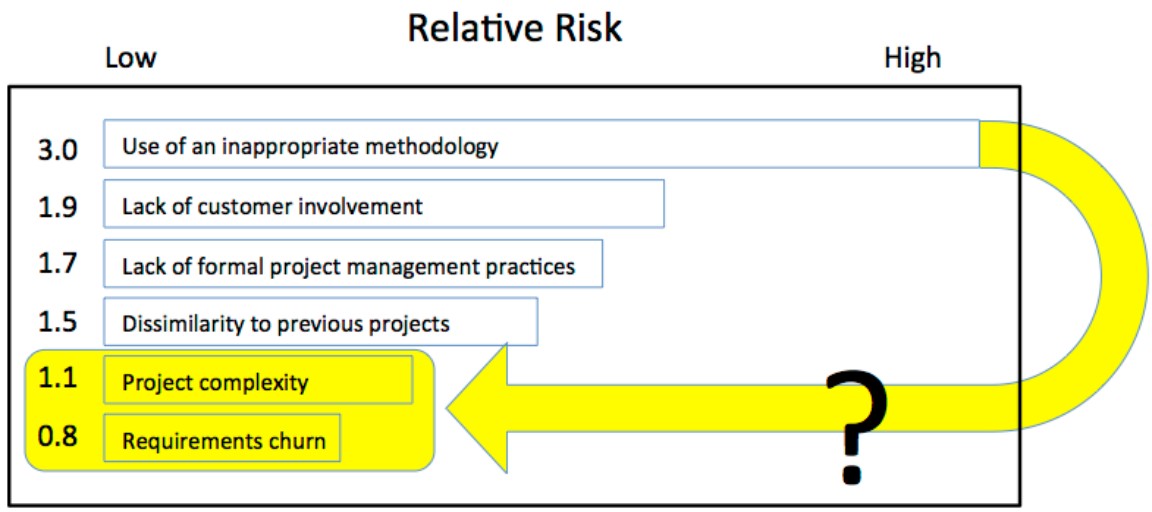
The first tool is called "Measure and Match," a procedure for evaluating how well a project management approach matches the management challenges of the target execution environment. Figures 0 through 6 present the core, with Figure 7 suggesting an additional use for the required measurements.

The second is the Interaction/Coupling Map, or "IC-Map." This is a tool to help facilitate conversations about options one may have to change the manageability characteristics of a system or environment. The associated diagrams are Figures 8 and 9.

Thank you for your interest in this work.

John

http://www.linkedin.com/in/johnlhelm/
This file: http://jhelmassociates.com/resources.html?item=MeasMatch
Version 1.1.5   2014-03-27



Tiwana and Keil (2004). "The One-Minute Risk Assessment Tool". *Communications of the ACM,* 47(11) 73-77.

Management approach mismatch ➡ Compromised risk management

Figure 0 – Method Mismatch Was Identified As the Biggest Risk in Study of 720 IT Projects

IT project frustrations have been part of our landscape for almost as long as computers themselves. The first conference addressing “software failure” and the “software crisis” was in 1968[1]. While the computing environments have changed a lot since then, the list of prescriptions to fix the problem has been remarkably stable. Although this discussion emphasizes IT projects, the results are applicable to projects in general.

An academic study of relative risk in IT projects by Tiwana and Keil[2] found that using the wrong project management approach presented the highest relative risk to project success. They were somewhat surprised to see that requirements volatility and project complexity were far less threatening than most other project management studies report. This result suggested that perhaps the sensitivity of a project to these risks was not necessarily an independent factor, but instead depended on how well they were addressed by the project management method used.

Since risk management is a key goal of project management, every project management method is premised on (often tacit) assumptions about the nature of risk. Tiwana and Keil’s results ironically suggest a significant and yet underappreciated “Mismatch Risk” exists; which is the risk using a management approach in a execution environment that is not well matched to the approach’s risk assumptions.

---

[1] *Software Engineering: A Report on a conference sponsored by the NATO Science Committee.* Garmisch, Germany. 07-11 Oct 1968. Peter Naur and Brian Randell, Eds.
[2] Tiwana and Keil (2004). “The One-Minute Risk Assessment Tool.” *Communications of the ACM*, **47**(11) 73-77.

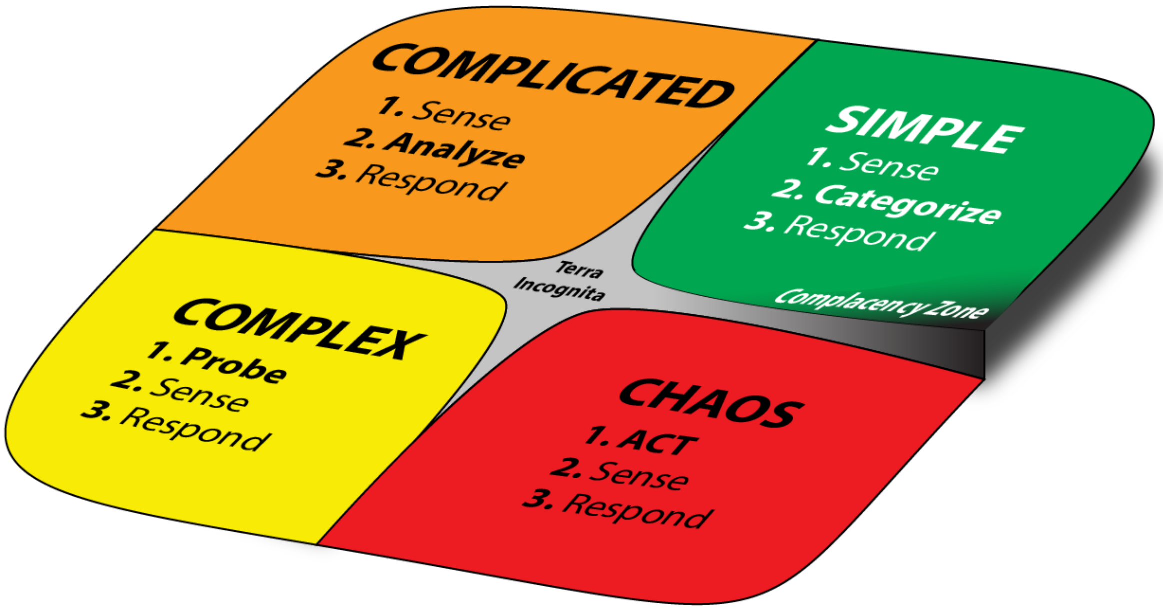


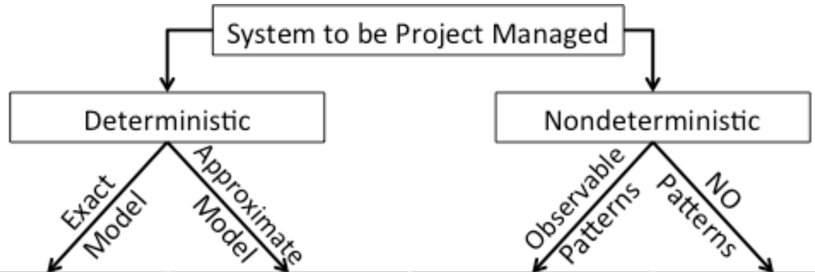
Figure 1 – A Visualization Of The Cynefin Framework

The existence of Mismatch Risk means there must be more than one kind of execution environment. Since all planning and management is premised on some notion of cause and effect, we'll use the Cynefin framework[3] for our classification system.

Cynefin framework defines four operating contexts that differ by how one experiences the nature of cause and effect. Simple and Complicated contexts are *deterministic*, meaning reliable cause and effect relationships are available for fact-based decision making. Complex and Chaotic contexts are *nondeterministic*, which means cause-effect relationships—to the extent they may exist—are *NOT* reliable. Nondeterministic environments are "Einstein Insane" in that the same input is not guaranteed to produce the same output every time. The difference between Chaotic and Complex contexts is in the former cause and effect cannot even be discerned, whereas in the latter patterns exist that are *reliable enough* to direct action. Large-scale weather is a good example of a complex system with reliable patterns. For example, "Red sky at night, sailors' delight; red sky in morning, sailors take warning" summarizes a very reliable, but not perfect, pattern for predicting rain within a few days. The capricious twisting and hopping of a tornado funnel provides a good example of an unpredictable chaotic system. Because any kind of premeditated action is impossible in the Chaotic context, it's not relevant to project management. The three-word sequences shown in the four Cynefin contexts are "action prototypes" recommended for decision-making. The grey region in the center is labeled "Terra Incognita" or Unknown Land; it corresponds to the state of not knowing one's context. This is where one has to be in order to be liable for Mismatch Risk.

---

[3] David J. Snowden and Mary E. Boone (2007). A Leader's Framework for Decision Making. *Harvard Business Review* (November 2077), Reprint R0711C.



	Simple	Complicated	Complex	Chaotic
PM Paradigm	Aim and Shoot	Analyze and plan	Iterate and adapt	"Follow Moses"
Examples	Procedure Book Waterfall Linear RUP Linear SDLC	Spiral Iterative RUP PMI/PMBOK PRINCE2	Scrum, Kanban, XP, dX, ASD, FDD, Crystal DSDM	Lean Startup The Prince (Open Source?)

Figure 2 – Example Cynefin Matched Project Management Methods

This figure presents a classification of selected project management approaches according to the best corresponding Cynefin action prototype. A corresponding Project Management Paradigm is offered since there is some project brand ambiguity out in the trenches of execution.

Aim and Shoot – work approaches that can be planned exactly and in their entirety before execution as a single sequence of work. Fast food restaurants work this way; place your order and after a very predictable interval, the entire "meal" comes out.

Analyze and Plan – project approaches that, if one works hard enough, can be planned with adequate confidence prior to resource commitment. This involves assessments or expert analyses of the known-unknowns. We do this in full service restaurants when we analyze the menu, plan most or all of the meal at once, and commit to execution by ordering it. Once done, each course comes out in turn.

Iterate and Adapt – work approaches that do not depend on planning an entire project before execution, because long-term planning is not reliable (by definition, because of nondeterminism). In the deterministic world, risk management is accomplished by assessment prior to commitment. In the nondeterministic world, however, one never knows when a pattern-based decision will fail. Thus, *risk is minimized by decomposing a project into as many small tasks as possible so as to minimize the value at risk at any given time*. This approach also allows subsequent tasks to benefit from lessons learned. Most people do this when faced with buffet of unfamiliar food. Instead of committing to a full plate, small samples are collected and tasted. The learning from this leads to getting more of what's liked and, perhaps, some additional samples—we iterate until full.

Follow Moses – this is the land of "Just Do it." Food-wise: you're shipwrecked on an island with no idea what is safe to eat. You can starve, or try something and maybe prevail…

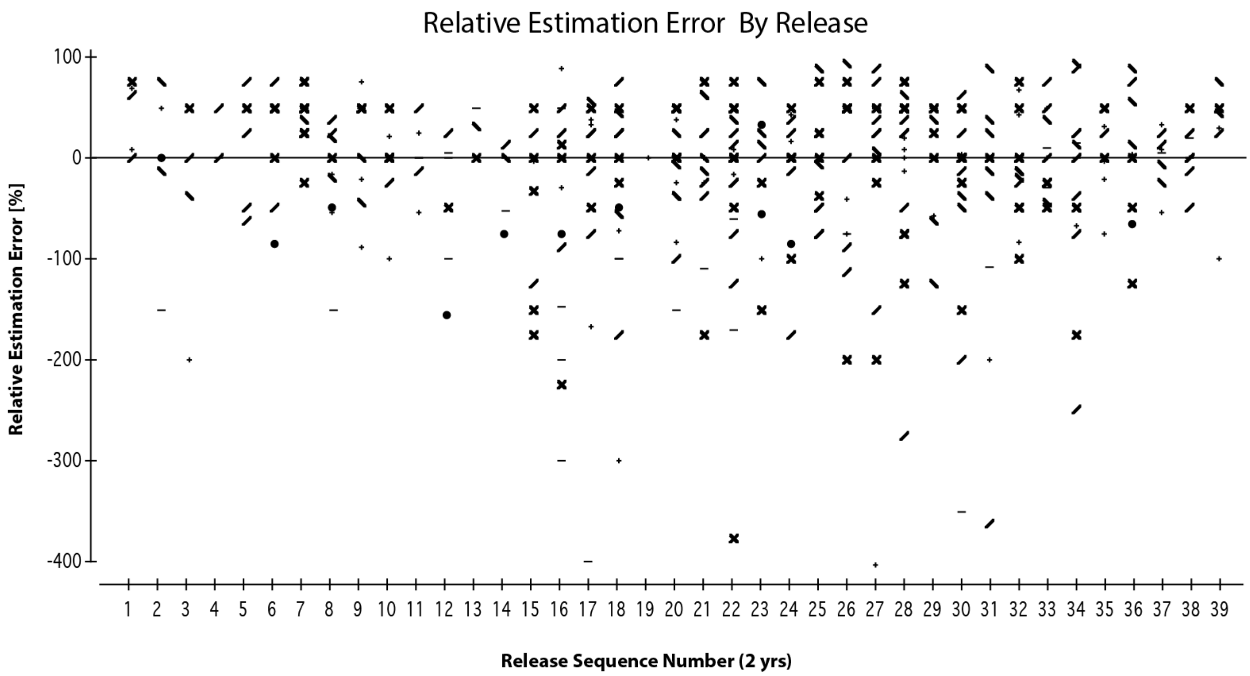


Figure 3 - Relative Estimation Error For 465 Implemented Software Features (Scrum)

So, how do we figure out what Cynefin context we're in? (This is called "sense making.")

Since every estimate of a project task, use case scenario or user story invokes knowledge of cause and effect, tracking the reliability of our estimates offers a way to quantify our experience of cause and effect.[4] We do this by tracking relative estimation error, which is given by (estimate – actual)/estimate. We use relative estimation error so we can compare estimates from tasks of different sizes.

Note relative estimation error will not be useful if we are not virtuous in our intent. The estimates needed are our best effort quantification of how much money or full-time hours of effort a task requires. This is distinct from *forecasting* which seeks to predict the date a task will be done, accounting for level of effort, delivery schedules, etc… Moreover, we assume the absence of coercive conditions that compel the analyst to bias. For example, in a culture where estimates are the same as *commitments,* what sane analyst wouldn't indulge in the bias called padding? To be helpful, we must track our estimation accuracy in good faith—otherwise, since metrics motivate behavior, it's better to abstain. Said differently, Measure and Match assumes a culture that believes "the truth will set you free."

---

[4] We note that accurate estimation involves several assumptions, including: a stable and testable description of "done" (requirements), mastery of practices and a comprehensive understanding of cause and effect (however encoded). For now, assume requirements stability is a property of the context and execution mastery is adequate. This topic will be revisited shortly.

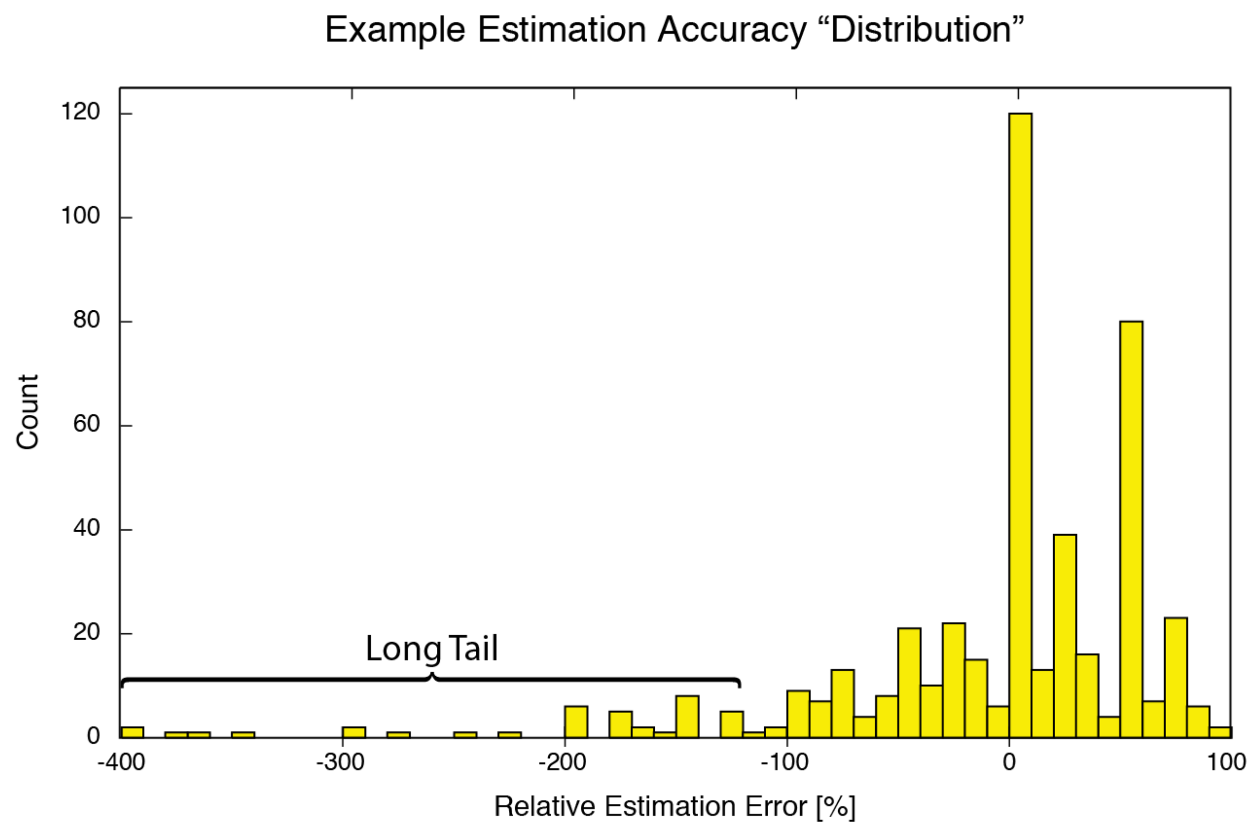


Figure 4 - A Relative Estimation Error Distribution With A Long Tail

The information in Figure 3 is not easy to interpret. To make it more so, we summarize it in a histogram. It may seem a bit unobvious at first, but all we're doing is taking the y-axis of Figure 3 and making that the x-axis here. For the y-axis on this plot, we show the count—i.e. how many times—a relative error of a particular size occurred. For example, careful examination of Figure 3 shows that a relative estimation error of -400% occurred twice, which is shown as a bar of height 2 on the left of this diagram. As there were no errors of -390%, the bar height is 0—that is, there is no bar; the next bar shows one error estimate of -380%, and so on. Redrawing our error history this way lets us obtain an empirical distribution of how accurate our estimates are over many attempts.

Note the "Long Tail" annotation on this graph. Long tails are possessed by distributions that have many extreme possibilities. A related term is "Black Swan"; where Black Swans are unpredictable or unforeseen events with severe consequences. One can think of Black Swans as "living" in the long tails. Recent findings by Flyvbjerg and Budzier[5] show over 16% of projects have very large black swan overruns. Clearly, when it comes to projects, unpredictable does not imply rare. Black swans in projects are significant off-balance-sheet liabilities. They manifest as hundreds of billions of wasted dollars per year; and worse, as newsworthy company failures such as Knight Capital or FoxMeyer Drugs.

[5] Flyvbjerg and Budzier (2011). "Why Your IT Project May Be Risker Than You Think". *Harvard Business Review*, September 2011, Reprint F1109A.

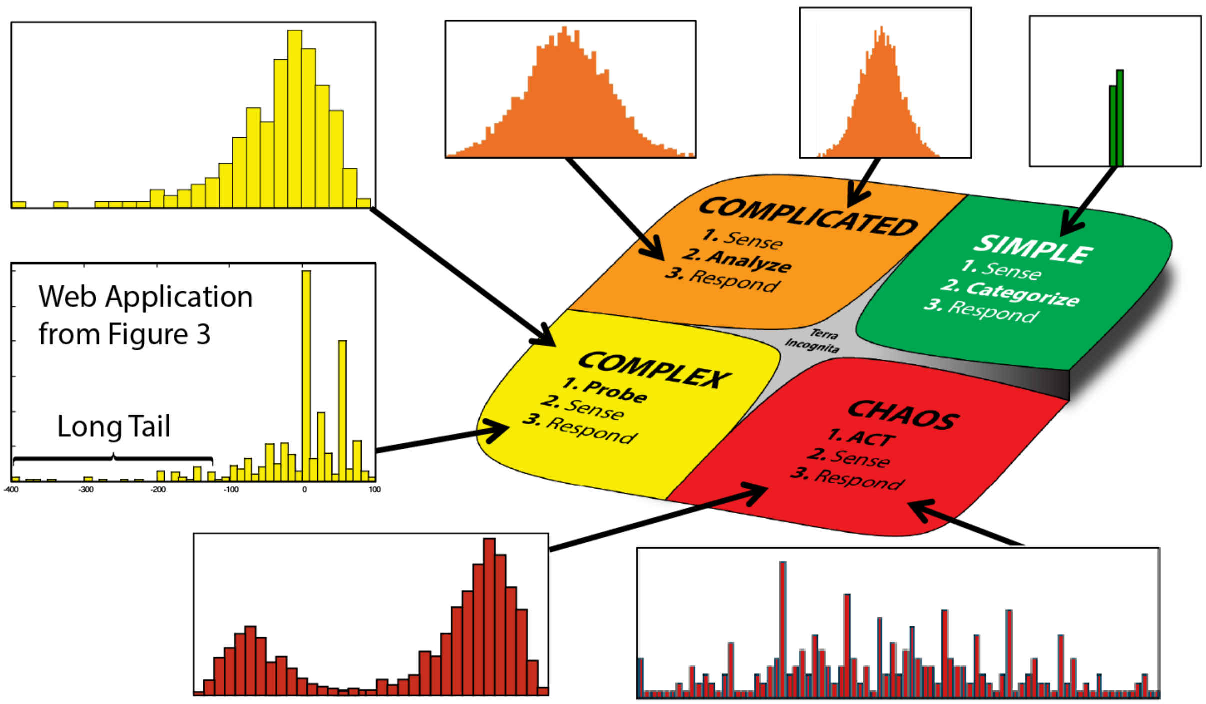


Figure 5 – Example Relative Error Distributions And Their Corresponding Cynefin Context

Here we connect the shape (or "morphology") of common relative error histograms to their corresponding Cynefin context.

Simple contexts are managed by "sense-**categorize**-respond" because there is very little variability as shown in the corresponding histogram. Poor results in these systems arise from execution mistakes in aiming (sense and categorize) or in shooting (respond).

Estimation in the Complicated context is more difficult but still useful if the result is stable over time (assuming the error bar can be made small enough). The "statistical determinism" of complicated systems gives process improvement approaches, such as statistical quality control, their efficacy. The action prototype here is "sense-**analyze**-respond," where, unlike categorization, analysis implies approximations are required.

The "simplest" manifestation of complexity is when the distribution is known, but the nature of that distribution makes planners sad. The unlabeled yellow histogram in Figure 5 is a "log-normal distribution." Skipping the math, what's important here is we know exactly what the underlying process is—but that's not all that helpful because the underlying process gives rise to a long tail. That means knowing the process does not equate to being able to plan with adequate efficacy. Here, prudence indicates we "**probe**-sense-respond" with small, risk-limited units of work.

The distributions in red give two examples of an estimation accuracy experience for which planning is not possible. Note the histogram with two peaks is an example of two superimposed distributions, so it's all bets off when it comes to computing statistics.

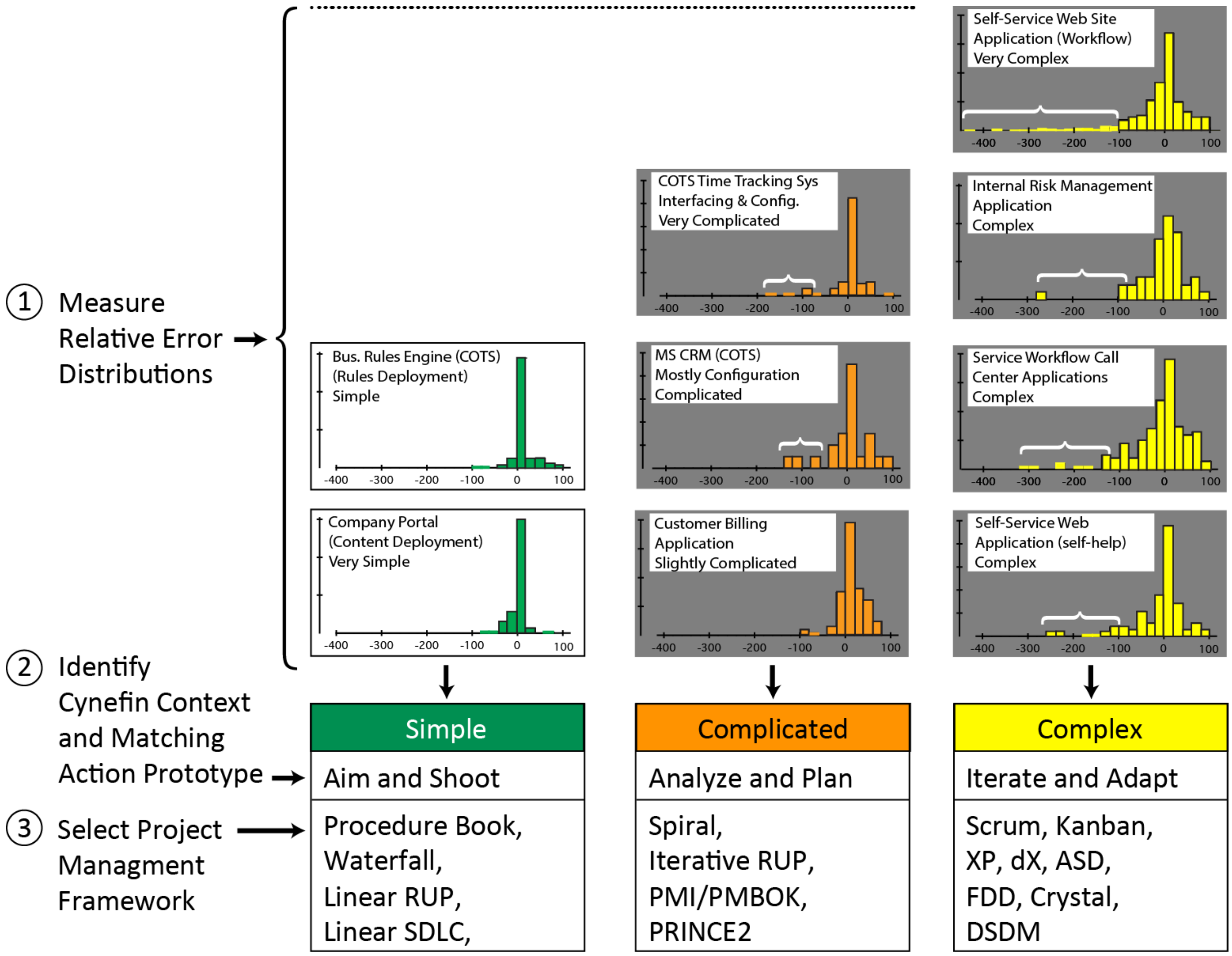


Figure 6 - Predictability Assessments Across An IT Portfolio

Here's an example of several codebases in a very large company. The relative error distributions shown were compiled over one year of monthly releases. It is very clear that these systems (code bases) are not all the same. The opportunity we have here is to match the most appropriate project management approach to each system on the basis of the measured estimation accuracy distribution. If all these systems where managed using the simple project management framework (Aim and Shoot), then several of these projects would be challenged and the risk of failure for the complex ones would be high. Conversely, if all projects were managed using a complex project management framework (Iterate and Adapt), then the simple ones would incur gratuitous overhead and therefore expense.

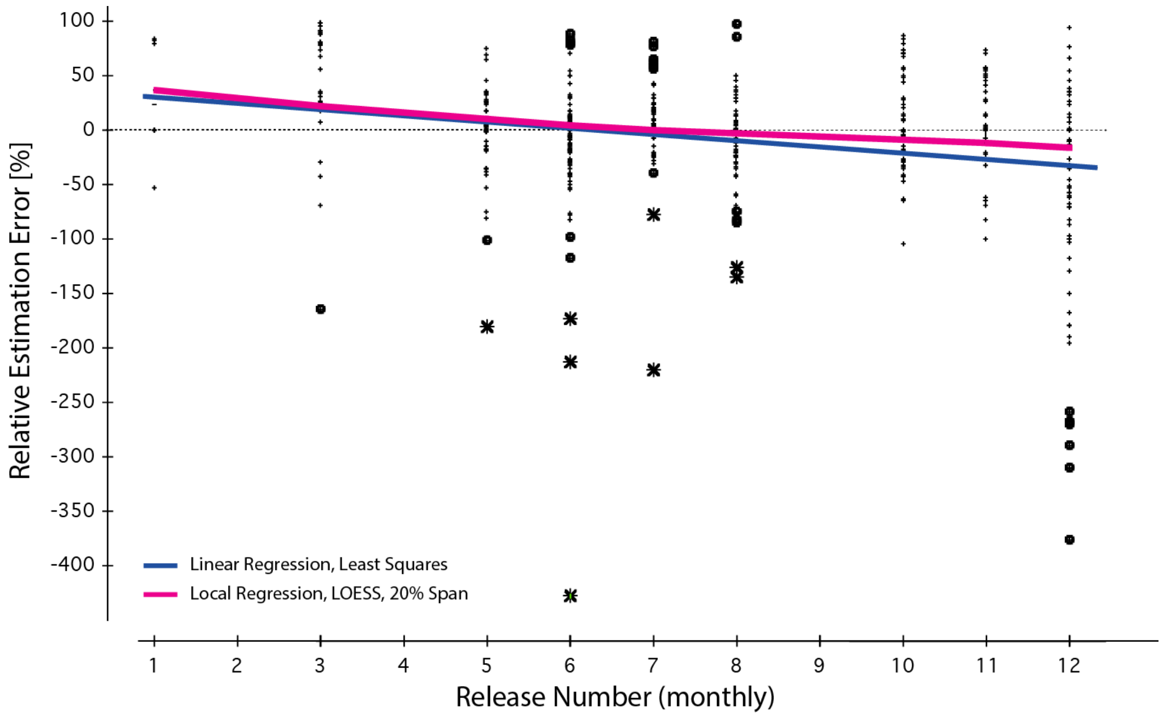


Figure 7 - A Decent Into Ever Increasing Complexity

Here we show how tracking estimation accuracy over time can reveal trends. In this case the trend is getting worse in the sense that estimation is getting less precise over time (i.e. the plus and minus range is getting larger) and the overall performance is trending towards increasing underestimation.

Optional Technical Notes

Curve fitting is not the same as statistics! When dealing with potentially pathological data, one must be forever vigilant about verifying assumptions. Note there are two trend lines; the blue line is a least-squares fit of a line whereas the red graph is a locally weighted non-parametric regression (LOESS). Without digging into the details, the two trending methods could not be more different. Thus, if these two plots differ in conspicuous ways, one cannot place much confidence in using a linear relationship to understand these data. Sadly, here the trend here is as clear as it is undesirable.

Note also that some points are in bold; using Tukey Box Plots (not shown), these are identified as potentially significant outliers. Again, without digging into the details, to a trained analyst the number and spread of these outliers screams "Pathological data!!!" Moreover, the conditions for convergence theorems, such as the Central Limit Theorem, are not satisfied; here, as observed by the singer-song writer Bruce Cockburn, "The problem with Normal is it only gets worse!"

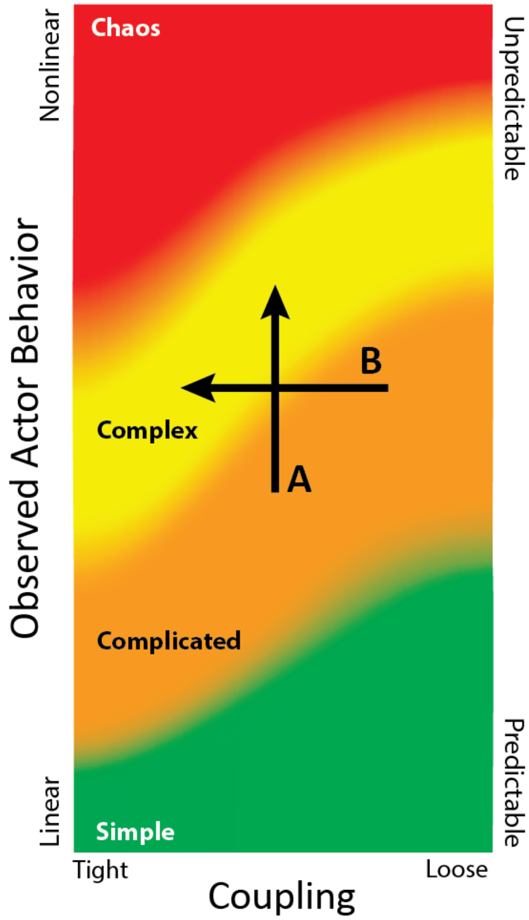


Figure 8 - An Interaction/Coupling Map Colored To Indicate Cynefin Contexts

Charles Perrow developed Normal Accident Theory to explain decision failures in complicated organizations. He examined how hard systems were for humans to understand based on two structural properties: component (or "Actor") complexity (simple/linear versus complex/nonlinear) and coupling (loose or tight). Perrow argues that tightly coupled nonlinear actors can interact in unexpected ways, overwhelming the decision makers involved. Figure 8 shows an adaptation of Perrow's Interaction/Coupling chart on which the four Cynefin contexts are indicated. The arrow marked A shows increasing complexity due to escalating non-linearity and arrow B shows increasing complexity arising from tightening coupling. This shows how a system can move in and out of complexity as load changes coupling tightness; just like traffic does before, during and after rush hour. Moreover, it makes clear we have only two choices to reduce system complexity: reduce coupling or, refactor or replace non-linear actors with more linear ones.

The actor-coupling model is very general in that most systems of practical interest can be represented as actors (computers, process steps or organizational units) that are coupled through flows, which can be information or physical things like parts. At a more abstract level, projects can also be interpreted this way; especially when viewed through the lens of critical path or network diagrams.

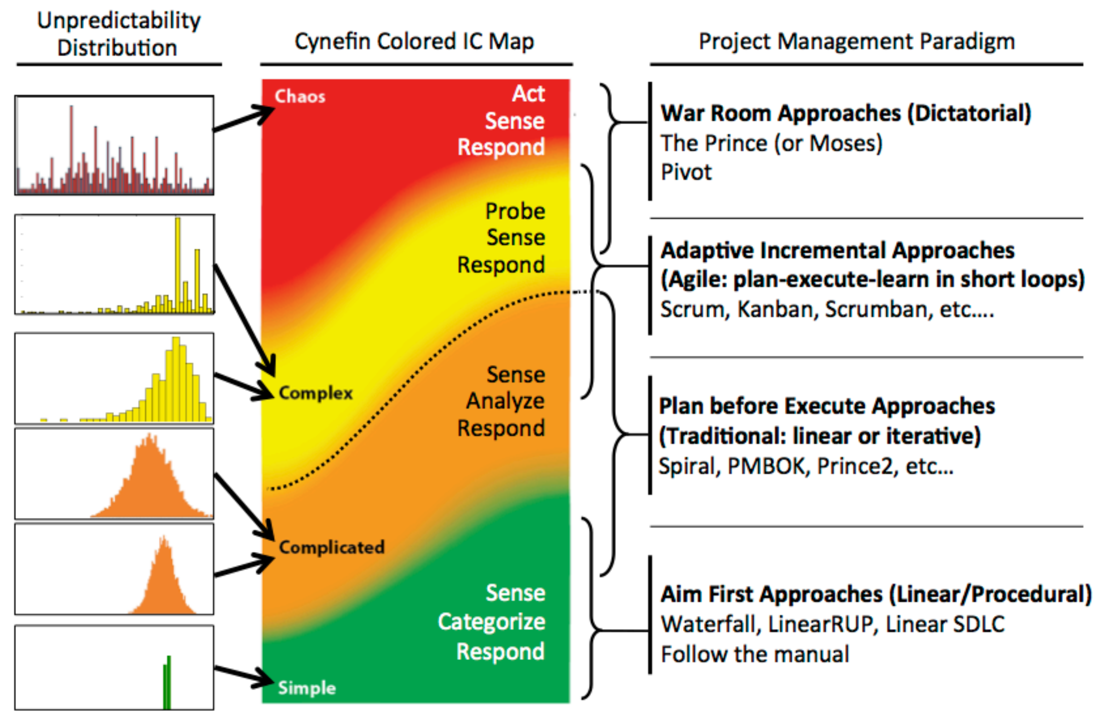


Figure 9 – Contexts Of Project Management Applicability On The Interaction/Coupling Map

Here we connect Measure and Match to the Cynefin-colored interaction/coupling diagram. Perhaps the most important observation is the sloped nature of the Cynefin boundaries. This means systems near these boundaries can behave ambiguously, or worse, inexplicably display a Multiple Personality Disorder, for example, as s function of load. Such equivocal signals often invite rationalization or outright denial. The best defense is to reduce one's dependency on subjectivity by measuring estimation error as much as practicable (continuously, if possible) so as to get a fully representative predictability distribution.

A final warning: analysts must be extremely skeptical of Classical (or frequentist) statistical analysis; more often than not the prerequisite assumptions are not satisfied. The real world of projects is very, very messy; often the measured histogram is a mash-up of more than one underlying process and almost always there is no way to "deconvolve" their signals. Even when there is a single, known process, if there is a "long tail" the behavioral result is still complex (e.g. if there is a log-normal or extreme value distribution). Indeed, while out of scope for this discussion, it's not hard to show that one should expect log-normal estimation error distributions for projects that involve a sequence of many dependent steps, each of which has a non-zero probability of failure.

Let's close where we began: requirements stability. Very often the shelf life of requirements is shorter than the time it takes to implement them. This volatility of purpose means the execution environment is complex even if the target system is only complicated (or simple!). Anyone who has remodeled a kitchen has experienced this. A key reason why Measure and Match is so effective is estimation accuracy captures requirements volatility. Indeed, estimation accuracy is independent of how the estimate comes true—this makes it an effective tool for evaluating the benefit of changing practices.