Epic Fails and Big Wins: Lessons from 20 Years of Data Infrastructure

Karen I. Stocks

Scripps Institution of Oceanography at the University of California San Diego

PRESENTED AT:

INTRODUCTION

Building effective operational data infrastructure is not a solved problem, it is an ongoing area of research. Fostering a culture where failures and pain points are openly discussed is as important for progress as celebrating successes.

I have worked on a suite of data infrastructure projects, ranging from small to large and simple to complex. Some have been successful, others less so, and a couple are so notorious they are generally not mentioned in polite company.

Looking across these projects, I consider both the technical and non-technical aspects that contributed to their successes as well as their shortcomings.

OPEN

LESSONS LEARNED THE HARD WAY: THE GOOD

The Good

Leveraging widely adopted tools vs developing home-grown solutions. When SeamountsOnline had to transition away from Oracle, there were tools to support this well-traveled path. More idiosyncratic systems are hard to maintain and hard to migrate.

Relying on external controlled vocabularies. CCHDO is transitioning to NetCDF-CF, in part to eliminate the time consuming work of developing community consensus around parameter naming. R2R minimized local vocabularies from the beginning, saving time and increasing interoperability.

Quality ongoing feedback from representative users (see box).

And common to **every** successful project I have worked on are:

- a focused mission with concrete benefits to a defined user community; and
- a devoted, invested staff

This poster is dedicated to the exceptional staff I have the privilege to work with now (Andrew Barna, Carolina Berys-Gonzales, Dru Clark, Steve Diggs, Sharon Escher, Jerry Kappa, Lynne Merchant, Chris Olson, Alan Yang, and the R2R team); and exceptional students current and past from whom I have learned as much as I have taught (Avery Ardent, Meagan Cummings, Renata Ferreira, and Anh Pham).

OPEN

LESSONS LEARNED THE HARD WAY: THE BAD AND THE UGLY

The Bad

The "publication paradigm": SSDB assumed that once data were submitted and made public, they would not be changed, but errors and other revisions were inevitable. The system was built on static text metadata files, requiring substantial time to remove in favor of a more flexible metadata database approach. CCHDO is wrestling with the same issue now.

Short term fixes for long term problems. One project's short-term decision to replace a failing system with a quick hybrid solution has created a long-term sustainability and functionality challenge, and a much higher price for replacement than a well-planned solution up front.

The Ugly

The "ugly" were serious enough, my opinion, to contribute to a project's downfall. They were unrecoverable errors for the sytems involved.

Inflexible devotion to requirements documents and initial planning over common sense. It is possible to create a system that passes every engineering requirement while still failing to be usable.

Lack of hardening and usability assessment. Several projects had a "more is better" mentality, adding new features before testing and hardening the previous functionality. Scientists didn't adopt these tools. Usability is an art and a science essential to all successful operational systems.

Hubris. One project devoted to being radically new and different ignored existing technology components and iterative improvements, running out of time to reach its potential

Lack of succession planning. When the SeamountsOnline funding ended, I kept it the site running without clear plans, and it eventually (and inevitably) aged and broke. In contrast, OBIS invested substantial time in developing a succession path, and is OBIS still thriving under IODE.

Interestingly, I have never been involved with a project that failed because it picked the wrong technology. It can cause pain and consume resources, but I have not seen a terminal impact.

OPEN

YOUR LESSONS

I asked attendees what their hardest learned or most important lesson was. Here are their answers:

- Asking scientists to use a tool that wasn't well designed and hard to use
- Done is better than perfect
- Real and perceived barriers to re-use; the "build it again" mindset. Nobody gets a pubication from reusing an existing tool.
- Having scientists code their algorithms vs. describe them in documents for the developers to code. It's faster to clean up scientists code than to go through document development cycles.

OPEN

ABSTRACT

Over time, I have worked on a suite of data infrastructure projects, ranging from small to large and simple to complex. Some have been successful, others less so, and a couple are so notorious they are generally not mentioned in polite company. Sometimes the outcome was predictable—well-organized and well-managed projects are likely to succeed, and those with clear flaws often don't rise above them—but sometimes it was unexpected. Building effective data infrastructure is not a solved problem, it is an area of research in and of itself, and the larger the project, the more difficult and uncertain it is.
For each of my current and past projects, I consider their greatest strength, their showcase best practice, their weaknesses, and their epic fails, as well as the external and internal factors that contributed to positive or negative outcomes. Despite the heterogeneity, certain patterns emerge as lessons learned: • Listening to your users is critical, and there are no short-cuts: it requires a long-term investment, including cultivating individual relationships.
• Good is better than more: hardening infrastructure is time consuming but critical for adoption. Have a clear mission with concrete benefits to a defined user community, and expand thoughtfully from there.
• In spite of our best intentions, meeting emerging community expectations usually requires a catalyst. External groups identifying best practices, mini-grants for implementation, and multi-project groups collaborating to rise together give needed nudges.

Interestingly, I have never been involved with a project that failed because it picked the wrong technology. It can cause pain and consume resources, but I have not seen a terminal impact.
Finally, invest in the people behind the infrastructure. Committed, engaged staff supported by ongoing professional development, rational management, and sufficient autonomy can, and regularly do, accomplish the impossible.

Over time, I have worked on a suite of data infrastructure projects, ranging from small to large and simple to complex. Some have been successful, others less so, and a couple are so notorious they are generally not mentioned in polite company. Sometimes the outcome was predictable—well-organized and well-managed projects are likely to succeed, and those with clear flaws often don't rise above them—but sometimes it was unexpected. Building effective data infrastructure is not a solved problem, it is an area of research in and of itself, and the larger the project, the more difficult and uncertain it is.
For each of my current and past projects, I consider their greatest strength, their showcase best practice, their weaknesses, and their epic fails, as well as the external and internal factors that contributed to positive or negative outcomes. Despite the heterogeneity, certain patterns emerge as lessons learned: • Listening to your users is critical, and there are no short-cuts: it requires a long-term investment, including cultivating individual relationships.
• Good is better than more: hardening infrastructure is time consuming but critical for adoption. Have a clear mission with concrete benefits to a defined user community, and expand thoughtfully from there.
• In spite of our best intentions, meeting emerging community expectations usually requires a catalyst. External groups identifying best practices, mini-grants for implementation, and multi-project groups collaborating to rise together give needed nudges.

Interestingly, I have never been involved with a project that failed because it picked the wrong technology. It can cause pain and consume resources, but I have not seen a terminal impact.
Finally, invest in the people behind the infrastructure. Committed, engaged staff supported by ongoing professional development, rational management, and sufficient autonomy can, and regularly do, accomplish the impossible.

REFERENCES

CLICK TO ENTER CONTENT

CLICK TO ENTER CONTENT