THE F-WORD

FAILURE IN INTERNATIONAL DEVELOPMENT
CREATING SPACE FOR LEARNING AND INNOVATION

By
Michael Eddy

A four-part series of blog posts
The F-word: Failure. It is development agencies’ worst nightmare and surely sends shudders down the spine of any program manager. One misstep and failure can kill entire careers or even organizations.

Through this series of blog-posts I am going to argue that development organizations as a whole are indeed failing. They are slow to learn, they are slow to innovate and they are slow to adapt. But, their biggest failure is a failure to know how to fail. That’s right, development organizations are failing because they don’t know how to fail.

The question is not will development agencies fail, but how will they fail and will they learn. A culture more open to failure is needed in development today in order to drive innovation and keep agencies agile.

Over the following two weeks, I’ll be posting a four-part series of blog posts addressing the following issues:

1. Failure and Experimentation
2. The value of failure: Why do development agencies need to fail differently?
3. How to fail well: Failing cheaper and earlier
4. Creating the space for failure

**Failure & Experimentation**

The approach to failure I lay out fits into a broader push towards greater experimentation. This approach differs from the academic exercise of “experiments” (namely, randomized experiments), which serve their own role separate from “experimentation”. Prichett (2012) most clearly articulates this distinction: “The experimentation approach is primarily an approach to practice. The experiments approach is primarily an approach to academic research.” The “experimentation agenda” has been by a number of different authors under a variety of different names. From “High bandwidth policymaking” (Hausmann 2006) to “good enough governance” (Grindle 2010) to “panners and searchers” (Easterly 2008). Pritchett (2012) has a more comprehensive list.

Nevertheless, the experimentation agenda is insufficient. For experimentation to be truly effective it must actively acknowledge and support “good types” of failure, while designing experimentation to avoid “bad types” of failure (as I will discuss in Part 1). Furthermore, without actively acknowledging the role that failure plays (and creating the political space for acceptable failure within the broader institutional space of development agencies),...
the experimentation agenda is bound to be another development failure (of the bad kind; see part 3).

Failure is inevitable. The next four blog posts will take you on a journey through the ups and downs of failure. The goal is to encourage development practitioners to think and talk more explicitly about failure. Throughout, I'll show how public, non-profit and private organizations have learned to use failure to maintain agile, high-performing programs in constantly changing environments. Each of these examples holds lessons for how development agencies operate their own programs and treat failure.
THE F-WORD: THE VALUE OF FAILURE FOR DEVELOPMENT AGENCIES
(PART 2 OF 4)

“We must also learn from failures, so that change will consist of the generation of organizational trials and the elimination of organizational errors.”

(North 1990, p. 81).

Organizations at the cutting edge of innovation understand the value of failure. The America’s Cup competitive sailing team, Team New Zealand, was able to out-perform other teams with much better resources by learning how to reduce the costs of failure using computer simulations. When Jeff Immelt became CEO at GE, he adjusted performance reviews in order to encourage risk-taking and create the space for failure, rather than just rewarding flawless execution of minimal-risk activities. Each of these examples show how failure can actively improve the way development is done today by enhancing **program-level learning** and supporting **organizational agility**.

- Failure is important because it enables **program-level learning and innovation** on what works and what does not in site-specific contexts. Creating an environment conducive to failure is most likely to be of greatest value when the program interacts with a wide-set of local and contextually-specific characteristics which is likely to influence how outcomes map onto the design space. This is because external evidence (drawn from dissimilar sites), in these circumstances, are not likely to be of much use in extrapolating. Instead, program parameters need to be tested in site-specific ways—where they interact with the full diversity of institutional constraints and behavioral norms. A good example is a hand-washing educational campaign. Effecting behavioral change requires interacting with a wide set of social institutions—how people perceive, and how open they are to different influencers. The results of a randomized experiment in rural Ghana are likely to be of limited use for a program in urban Mumbai. Getting to the field and rapidly testing and iterating is likely to generate much more program-relevant knowledge, in a way that is faster and cheaper.

On the other hand, a process of experimentation and failure is unlikely to be useful to test whether anti-retroviral drugs are effective at prolonging life among AIDS patients in Botswana—evidence from other sources (even US drug trails) are likely to hold strong claims to what might happen in Botswana. A better structure for understanding what types of evidence might hold stronger claims of external validity than other forms of evidence, is clearly needed.

- Failure is also important because it maintains **organizational agility**. Creating space for actively reporting failure helps development agencies to learn and to maintain an agile, high-performing portfolio of projects in organizations otherwise sluggish to recognize and move-on from failing programs.

A **Typology of failure**

Failure fails when it is used as an excuse to shirk responsibility or diffuse accountability. The inability of failure to gain traction as an acceptable way of learning stems, at least in part, from the lack of differentiation between good and bad types of failure. Amy Edmondson, a professor at Harvard Business School **describes three types of failure** (Edmondson 2011):

- **Preventable failures in predictable operations** – Like a bus driver failing to stop at a red light, this failure is unambiguously bad. Better processes and training to prevent this type of failure (and punish it when it does occur) are important for maintaining accountable
organizations. Financial management procedures and safety requirements, for example, would generally fall under this category.

- **Unavoidable failures in complex systems** – This type of failure results from the complexity of the environments in which some organizations operate. This type of failure may or may not be attributable to an individual. While not necessarily preventable, these types of failure can be managed in order to lower the costs of failure. For example, the design and launch of a NASA space shuttle is undoubtedly a complex system. Nevertheless, had NASA engineers done earlier experimentation and realized that a key gasket, called the O-Ring, fails in cold temperatures, failure could have occurred earlier on in the design cycle and the Challenger Shuttle Disaster may never have occurred.

- **Intelligent failures at the frontier** – This is the best type of failure. Many key questions about development policy are simply not knowable in advance: How will mothers react to being asked to pay a nominal fee for bed nets? Will households change their behavior by using a biogas digester in order to save money on charcoal and reduce indoor-air pollution? While this type of failure can still be made cheaper and earlier, failure associated with “trial and error” learning must be actively encouraged, not punished.

Creating an environment where development agencies are learning from failure means supporting intelligent failure, while actively discouraging and reprimanding predictable failures. Experimentation in development agencies is fundamental to allowing the best kinds of failure (those at the frontier of knowledge) to occur in real conditions. Now that we understand the value of failure and some different types of failure, next week I'll cover precisely how to fail better.
HOW TO FAIL WELL: FAILING CHEAPER AND EARLIER (PART 3 OF 4)

A Theory of Optimal Experimentation

The dominant approach to the evaluation of social programs (randomized evaluations) are excellent at cleanly identifying causal effects—but they often cost several hundred of thousands of dollars and take multiple years before the results are published. Failure is caught far too late in the design cycle and after far too much investment. The results simply take too long to feedback into the design process.

Figure 1 lays out a stylized theory of optimal experimentation. The marginal benefit to experimentation is declining (the next experiment probably produces less learning than the previous one). Cheaper experimentation, in this model, therefore enables social programs to do more experimentation and ultimately to learn more.

Of course, cheaper experimentation may also come with a tradeoff in terms of rigor, but as I explain below, this tradeoff does not necessarily have to be true: there are methods of cheap randomized experiments that can be employed to create internally valid, site-specific causal statements (e.g. in situ lab experiments). Even when there are indeed more explicit tradeoffs between cheaper, but less rigorous experimentation, this tradeoff is likely to produce gains when interventions are highly context-specific and (roughly) internally valid, local site-specific knowledge is more useful than translating “rigorous evidence” from other dissimilar sites.

What does cheaper and earlier failure look like? Cheaper and earlier failure in development agencies would encourage “good-enough” experimentation. Good enough experimentation attempts to isolate as many confounding variables as possible and prioritizes rapid prototyping and iteration over perfectly constructed counterfactuals in order to learn and innovate faster.

Team New Zealand, in testing out new shapes of their keel, decided to put two boats in the water and run them next to each other. This was not a truly randomized experiment—it failed to isolate all possible confounders—but it did isolate many of them. Racing time, combined with the team’s own intuitive reaction of how the boat felt, allowed Team New Zealand to narrow in on the best design and cut a full 2 minutes off their race time.

Similarly, when Intuit wanted to experiment with how to get their customers calling tech support to the right answer quicker and with less transfers, they constructed a “rapid prototype” of what the real thing might look like. Rather than re-engineer their call routing software (which would cost hundreds of thousands of dollars), they instead answered a few calls with a human technician who would read through the machine script. This “fake backend” wasn’t a perfect simulation of real circumstances, but it

---

1 One could imagine a more sophisticated model whereby probability of failure was a function of complexity of the problem (X) (i.e. adaptive vs. technical problem), and the openness of the organization to failure (O). Learning was a function of failure (f), validity (either externa validity or internal validity if it’s a site-specific evaluation) of evaluation method (v) and ruggedness of design space (r) and perhaps other variables. Failure imposed a cost c. Rigor was proportional to time to evaluate (t) and cost of the evaluation (ce). One could even think about setting different objective functions for practitioner versus academic. The (benevolent) practitioner would maximize learning and/or outcomes, the academic would optimize rigor.
did allow Inuit to rapidly adapt the script and re-configure the routing procedure as they learned what was working well and what was not.

Development agencies can do the same. Just like Team New Zealand, development agencies can pilot different mixes within the design space across their catchment area in slightly different ways. This might take the form of a cross-cut experiment which has the advantage of being able to see interaction effects of design parameters, but requires a large sample size. And just like Intuit, program design characteristics need not be the full-fledged intervention, but can be paired down to cheaper prototypes of the key operating components in order to enable rapid feedback. By comparing the data coming in and triangulating it with program manager’s own intuition about how different design features are working in the field, development agencies can actively update their prior beliefs about what is likely to work on a large scale (even if it’s not cleanly identified conclusions). This ensures that when failure inevitably does occur, it stays small, stays cheap, and occurs early in the design cycle.

Other approaches to keeping failure cheap and early include in situ lab experiments or simply evaluations based on outcomes (which are more likely to be realized in a shorter time-frame) rather than impacts.

**Strategically Crawling the Design Space.**

The policymaker can look across the design space (i.e. the multidimensional space of all design characteristics forming the set of all possible program designs), identifying a few key design parameters that have the potential to yield high outcomes (based, of course, on the international evidence, but also drawing from the *metis* of local actors—Scott 1998). Here, we may very well see a risk-return tradeoff. Programs that are likely to yield the best outcomes may also be the riskiest (if only by the fact that was it a high return-low risk design mix, the design space can be collapsed in that dimension). While different authors have different approaches to picking design mixes within the design space (Pritchett et. al 2011; Fraker, Shah 2011), one approach to maximize learning (by construct) and potentially maximize outcomes (if the risk-return tradeoff holds empirically), is to pick programs with the highest variance (i.e. uncertainty) in outcomes. Figure 2 presents an approach to doing this over the lifecycle of a project.

**Figure 3:**

*Actively crawling the design space by beginning with a wide variety of design mixes and narrowing down to optimal “best fit” design parameters.*

**Replicating in Situ Lab Experiments**

In-situ lab experiments are small scale, rapid simulations in the field done to test one specific design parameter or causal mechanism. I briefly give an example that is indicative of the relevance for development programs more broadly:

- **Does access to formal savings crowd out informal insurance?** One mechanism through which savings programs purport to achieve impact is by enabling households to better manage risks and be more resilient to shocks. However, if formal savings is simply crowding out informal insurance (which is already a well established characteristic of many communities), then formal savings accounts are likely to have much smaller impact.
Chandrasekhar, Kinnan and Larreguy (2010) test this by simulating multiple rounds of positive and negative shocks in a carefully controlled “lab experiment” conducted in 34 villages in Karnataka, India. In one treatment arm participants have access to intertemporal accounts to save across rounds. In another treatment arm, they are only able to smooth consumption in each round through inter-personal transfers with a partner. They find that savings does not crowd out insurance in this sample and have some interesting results on how social distance affects ex ante commitments to informally insure their partner. This result translates directly into how one might design a program providing access to savings accounts. 

Academics are seemingly allergic to replication studies due to the perverse incentives placed on them to publish new research. Since academics have shirked on their responsibility, policymakers must start adopting these tools. The intellectual capital has already been invested (by Habyarimana et. al) in designing the experiment. I would not expect policymakers to innovate in designing new experiments, nor testing new mechanisms. Rather, there could be tremendous value in simply replicating existing experiments in new contexts. The intellectual capital necessary for a replication would probably not rise to the level of a PhD (an MPA/ID would surely suffice!). There is tremendous program-improving value that could be achieved by conducting more replication studies.

**Unstructured Experimentation**

All of the approaches above have been structured experimentation—the policymaker enters ex ante with a design for testing key hypotheses. However, successful experimentation need not be so structured from the top-down. **Encouraging greater autonomy at a local level for innovation and potentially failure** can be just as effective at inducing a culture of experimentation that enables high-impact local innovations to be scaled. 

For example, the Charlotte School District, through a policy of “Freedom and Flexibility with Accountability,” allowed high performing principals to circumvent established policy and procedures as long as they continued to deliver high results. Some principles tried out single-gender classrooms; others put multiple teachers in one classroom; still others used new technology in the classroom. Complementing these local innovations was the systematic use of data which allowed Charlotte School District to better understand what was working best.

---

2 Another example of a program-design relevant question that could be answered through a lab experiment: **Does ethnic heterogeneity undermine the types and maintenance of public goods? How will this affect a public works or community driven development program?** A relatively easy approach to answer this question would be to replicate the Habyarimana et. al. (2007) study on ethnic diversity and public goods provision. The study carries out an excellent set of in-situ lab experiments to attempt to explain why ethnic heterogeneity leads to under-provision of public goods. Habyarimana tests three different families of causal mechanisms. Each one of these mechanisms might lead to a different set of design parameters in a public works program or a community driven development program (for example, what level of aggregation defines a “community” in a CDD program.). While results have strong internal validity, they are unlikely to have much external validity—the way that co-ethnics and non-co-ethnics interact in urban slums of Uganda is very likely to be different than the way they interact in the rural districts of South Sudan for example.
CREATING THE SPACE FOR FAILURE (PART 4 OF 4)

So if failure is so valuable, and we generally have some approaches to knowing how to fail cheaper and fail earlier, why are more development agencies not learning how to fail better? Is this an area where we can really translate private sector approaches to public sector organizations?

WHY PUBLIC SECTOR ORGANIZATIONS IN GENERAL (AND DEVELOPMENT AGENCIES IN PARTICULAR) ARE DIFFERENT

First, many argue that there is too much at stake in development agencies. Failure in the private sector may mean failing to make your quarterly numbers, or worse still, having to shut down a business unit. But failure in development might literally mean life or death (patients not getting the life-saving drugs they need) or a missed opportunity (a generation of kids failing to get access to quality education).

Second, the fact that social programs have multiple stakeholders means that outcomes are only occasionally the main focus. In business, there are clearly defined metrics: # of customers reached, revenue generated, etc. In development organizations, programs are often beholden to multiple different stakeholders—financers, taxpayers, political supporters and too often lastly, the end beneficiary. The absence of quality data and evidence often creates the space where anecdotes and political pressure take the place of better informed, evidence-based policy. Even with this evidence we still see the persistence of many failed development policies, such as tied aid and centralized state-planning.

Third, even when outcomes are the focus, internal incentives are not aligned. Agencies, departments, managers and front-line workers often are forced to absorb the downside risk of failure without capturing any of the upside benefits. This means that when things fail, front-line workers are often held directly accountable (i.e. losing their job or deferred promotions) More importantly, when good things happen, rewards are minimal. Rarely are they clearly articulated and when they articulated, they are not nearly as high as in the private sector. This creates an aversion to risk, ultimately choking the air out of any innovation in the organization. Innovation and learning is a two-sided coin: we must accept certain types of failure some of the time, in order to enable the real transformative change.

CREATING THE SPACE FOR FAILURE

Having understood the differences between public and private sector organizations I draw 3 lessons that might help public sector organizations better manage and encourage learning and adaptability through failure

1. Differentiating good failure from bad failure up front: Don’t kill the messenger & “Failure Audits”

Being clear ex ante about what types of failure are okay should help to create more political space in the organization for “good” failure. This begins by creating an environment where reporting failure is accepted and encouraged—don’t kill the messenger.

Performance metrics must be aligned to encourage good failure and to punish bad failure. After-the-fact “failure audits” can help get to the root cause of why failures occurred, take appropriate corrective action to the design parameters of the program and incorporate the learning more broadly back into the organization. The failure audit should be primarily a learning exercise not a disciplinary mechanism.
2. Managing expectations: Embedding failure inside larger portfolios

Another approach to creating the space and openness to failure is by explicitly setting targets on what percentage of programs (within a portfolio) you expect to “fail”. Actively searching for failures, and/or defining failure as performing at the bottom of the portfolio, helps organizations learn what design characteristics might not be working well, while also maintaining agile portfolios that weed out poorly performing programs. The limitations and potential concerns of this approach are non-trivial and I discuss them more in part 4.

3. Lowering the political costs of failure: “Failure redress” mechanisms

Well-designed social programs have grievance redress mechanisms—program designers know that program execution may not be perfect in all cases and build in established processes for addressing individual grievances. Likewise, I propose that the experimentation agenda, for it to be effective and politically supportable, should have "failure redress mechanisms". Despite a process of experimentation that encourages cheaper and earlier failure, failure will still, of course, occur. When and where it does occur individuals who bear private costs from failed programs should be adequately redressed and/or compensated. This compensation can come directly from the welfare rents from experimentation (in a sense, cross-subsidizing failure from the gains to experimentation). By establishing these failure redress mechanisms ex ante, not only is the program designer managing expectations and preparing the public for certain types of failure, but they are also lowering, if not removing all costs to failure—reducing the likelihood that individuals in failed treatment arms mobilize resistance to the program.
Bibliography


