

Getting there: A case study of REI's journey of getting the right data and environments for their enterprise testing ecosystem

Bob Stuart & James Wilson

bstuart@rei.com; jamwils@rei.com

Abstract

Like any retailer, REI is in an industry that changes frequently and increasingly leverages technology to differentiate itself and improve efficiency. The pressure to keep up with this is, like many companies, felt deeply within the QA department. REI QA found itself having a great staff and desire to help the company improve its delivery capability, and in 2015 REI showed its support for QA by hiring its first director of QA. Priority #1 was an improved QA practice, but it was clear that some combination of technology, environments and data needed to be dealt with. But how to get there from here?

In this case study you will read of REI QA's multi-year journey toward the lofty goals that so many aspire to: intentionally managing test data and test environments. You will learn of the reasons for change, projects that have been put into action, the reasoning and business justifications, lessons learned, organizational adjustments, achievements to date and what is next in the roadmap.

Biography

Bob Stuart is a QA System Engineering Lead and subject matter expert at REI with 10+ years of experience in IT test environment planning, scheduling, architecture and management. His career has covered the spectrum from hands-on desktop support, to server and network engineering, seven years in the SAP application space and, prior to QA, being a charter member of the ITSM team to launch Release and Change management for the REI enterprise. Bob has a mathematics degree from Seattle Pacific University, and multiple network engineering and IT process certifications.

James Wilson is a Sr. Project Manager at REI on the IT Enterprise QA team. He has enjoyed a career in IT and technology companies in a variety of roles, such as management consulting, project management, business systems analysis, and technical communications. Along the way he led SDLC adoptions, PMO lifecycle rollouts, agile and waterfall projects, and many vendor product evaluations. He successfully implemented adoption of new technologies and processes by applying some key principles: 'begin with an end in mind', 'light many small fires' to bring about big changes, don't be afraid to make mistakes, and never stop learning. He holds an MBA as well as a Masters in Management Information Systems, both from the George Washington University.

1 Introduction

REI QA was, for years, a department within IT that was largely an ‘order taker’ group and struggled to influence software delivery and project management. In 2015 QA hired its first director and was given a seat at the table for improving software delivery and project execution. Starting with that first year there were two parallel tracks of work that moved forward.

1.1 Assess and improve QA core practice

This effort consisted of a strategic hire of a QA lead with the experience and skills to consolidate, document and roll out a standardized set of practices and deliverables. This was executed well and gave a boost to the team in the form of standardized practices, process and document templates.

1.2 Test environment planning

The concern of this track was the quickly accelerating work starting up to ‘move to the cloud’. This began with a QA leadership alignment effort in which an approach to future test environments was designed, based largely on the static, on-prem environments of the past. (see *Figure 1*)

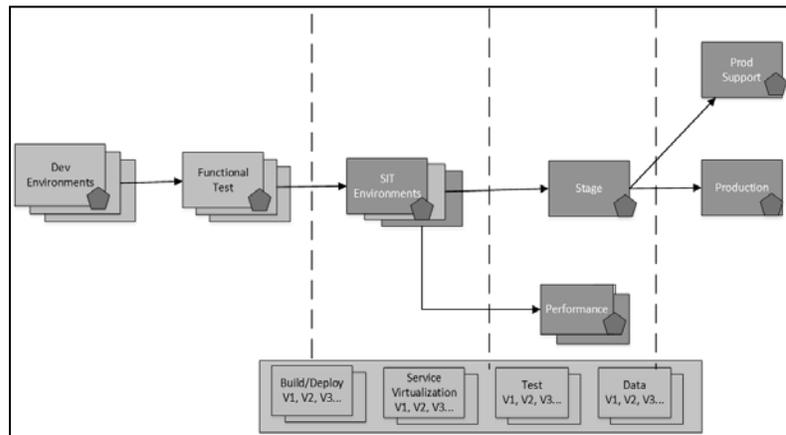


Figure 1

From the environment alignment effort (1.2) a project was sponsored to draw up the requirements---from a QA perspective---for a well-designed *hybrid test environment* (on-prem and cloud infrastructure). The end goal was to have a blueprint for test environments that could be executed when ready to migrate to the cloud. This effort was led by a niche consultant and documented each team’s capability and concerns with going to the cloud. The output of this project was shelved leading into 2017.

****Lesson learned #1**** Looking back, 2016 was full of cloud hype and our team was being [possibly overly-] proactive and taking on responsibility for planning things over which they ultimately had no control. The company was not ready to migrate to the cloud, so it was a matter of missed timing.

2 A good start – define the problem

Just as the hybrid test environments blueprint was completed, events led to the team being without a director for the last half of 2016. It was in this period that the team began to plan for the 2017 budgets and, in the absence of a hybrid environments project, the focus shifted to more textbook TDM (Test Data Management): refresh, synthetic data, fencing/reservation, etc. A number of meetings were held with vendors and there was cautious optimism that QA might be able to sponsor a TDM project.

In the vendor meetings it became clear that a business benefit was needed in order to move forward, so an effort was launched to assess the impact of test data issues on the QA team. The goal of the assessment was to develop a business case upon which a TDM project might be chartered. The assessment used a two-pronged approach, focusing on two areas:

2.1 Waste analysis

This was a survey of the team to discover how much time was spent across all portfolios to prepare and deal with test data. The average was approximately 12% of the QA team time, which was much less than the informally tossed around 25% industry average. This rings true, given the omission of the non-QA time (software developer, DBA, etc) spent on data manipulation. The dollar impact to QA per year was estimated to be ~\$1.2M of the team's time spent on non-value-add test data management.

2.2 Pain points survey

This was an anecdotal survey of the QA Team regarding any and all known data issues. The line items were organized into similar categories and rated on overall impact to the team. The consolidated pain points details are shown in *Table 1*

Top Issues that cause problems	Issue Count
Changed data	13
Data not sync'd	15
Data setup problems	7
Environment Availability	1
Environment Configuration	1
Environment Issues-Need more detail	4
Inability to identify data with correct attributes	1
Missing data	4
No QA support	1
Stale Data	1

Table 1

3 Assessing solutions

After attempting to define the problem and not delve into solutions, it was time to discuss what technical people love—solutions to problems. The solutions under consideration were a blend of classic “TDM” and more technical solutions. The following definitions were used by the team for this exercise

3.1 Solution coverage

In this stage of the process the team collectively assessed how well they felt each of the possible solutions might solve the various pain points. Each solution was discussed, and the relative merits or applicability to the problems was debated. The end result was a pivot-table driven mathematics exercise of lining up the problems and the solutions, with the resulting score representing how well the collection of solutions could solve each problem. It was felt that this score would indicate which solutions to pursue and in what order.

Term	Definition
Dynamic data	ad hoc or on-demand, at time of test, non-static data, disposable (could be programmatic); self-service
Synthetic data	planned or ad hoc, Fake/generated data, can be deterministic or random, new stores as example, based on seed data
Gold copy	planned, based on Prod usually, highly analyzed to ensure validity E2E, contract testing
Subsetting	planned, not full copy of db, culled/scrubbed for dupes or edge cases, can be sliced by time or process
Refresh	planned or ad hoc, copy from other instance (prod usually), can be full copy or tables, etc
Data fencing	planned, data reservation, protection of data so others can't use or change it
Service Virtualization	Full service logic, record & respond, generates dev artifacts/versioning, data can be managed in background
Mocks	Minimal logic, returns 'canned' response, may have limited exception handling or error checking
Stubs	No logic, returns 'canned' response, no exception handling or error checking
Db Virtualization	Minimal pointers that access full copy of data, all data with no subsetting
Masking	Can be deterministic or not; change fields to hide real member/customer data; must be done across core TDM systems concurrently
PII Data	Those personally identifiable information (PII) data elements determined by REI business owners and Enterprise Architecture that put customers and the company at risk

Table 2

Table 3 shows the final “Solution coverage across all problems” in the bottom row, which we viewed as the best indicator of what QA should focus on as solutions

Top Issues that cause problems	Issue Count	Solutions											Solution coverage by pain point
		Dynamic data	Synthetic data	Gold copy	Subsetting	Refresh	Data fencing	Service Virt.	Mocks	Stubs	Db Virtualization	Masking	
PII data	36	0	5	4	5	0	3	4	5	5	4	5	40
Data not sync'd	15	4	4	3	0	4	3	5	5	5	4	NA	37
Changed data	13	5	3	0	3	4	4	5	5	5	4	NA	38
Data setup problems	7	4	5	4	3	2	2	5	5	5	4	NA	39
Missing data	4	0	5	5	0	2	0	5	5	5	5	NA	32
Environment Issues	4	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	0
Inability to identify data with correct attributes	1	3	3	4	2	0	0	3	3	3	3	NA	24
Stale Data	1	0	4	4	0	4	0	3	3	3	5	NA	26
Environment Availability	1	NA	NA	NA	NA	NA	NA	5	3	3	NA	NA	11
Environment Config	1	NA	NA	NA	NA	NA	NA	3	3	3	NA	NA	9
No QA support/coverage	1	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	0
Solution coverage across all problems		16	29	24	13	16	12	38	37	37	29	5	

Table 3

4 Decision Factors

The team concluded the assessments as 2016 ended, just in time for a new director to take the helm of Enterprise QA. After reviewing the available information and options the decision was made in early 2017 to pursue PII data masking and database virtualization (DV). This decision was influenced by several factors:

4.1 Logical order of projects

Service virtualization (SV) had the highest scores and was seen as a big boost to overall software development and testing, yet ultimately needs a robust data plan behind it. There was not enough budget to pursue one of the enterprise TDM tools, so the team considered point #4.2

4.2 Pragmatism

The DV toolset was already purchased and stood up in a proof-of-concept fashion--and the QA team was welcomed to take this ball and run with it. Thankfully, the database team was a great partner in supporting QA's interest in establishing a robust foundation for getting data and environments in the hands of testers.

4.3 Risk management

The DV platform purchase included a robust data masking toolset, which got the attention of Enterprise Architecture. This was one of the key use cases/issues called out in the QA assessment (and in Architect interviews), so pursuing DV while simultaneously dealing with data security was seen as a huge win.

4.4 Strategic alignment

The new director came in with a vastly different perspective on test data and test environments and how they can help drive software development and testing--it was certainly different from the original team alignment of the previous year. This change in perspective resulted in significantly reducing the focus on infrastructure and cloud test environments, and focusing on providing core testing capabilities and capacity for the enterprise.

5 The North Star

5.1 Shifting of the tides

The new director arrived having previously seen the impact that targeted solutions can have on the QA practice and software development overall. Soon after arriving she worked with the QA engineering team to develop a target state for test environments--referred to as our "North Star", or "Polaris". This was a set of graphics and a slide deck showing multiple approaches to test environments for the cloud use case. This became necessary in order to help clarify the software testing/delivery approach being communicated throughout IT.

Figure 2 was created to illustrate how QA partners with other teams that already own or specialize in delivering infrastructure and providing systems needed for software delivery across the REI landscape. This illustration was very cloud oriented, reflecting the focus at that time, but easily maps to on-premise infrastructure.

The vision is that QA will sit alongside other teams in a shared VPC (virtual private cloud) and provision the many databases and services needed to conduct effective testing. This view of the future presumes a high degree of collaboration and alignment between QA and these other teams.

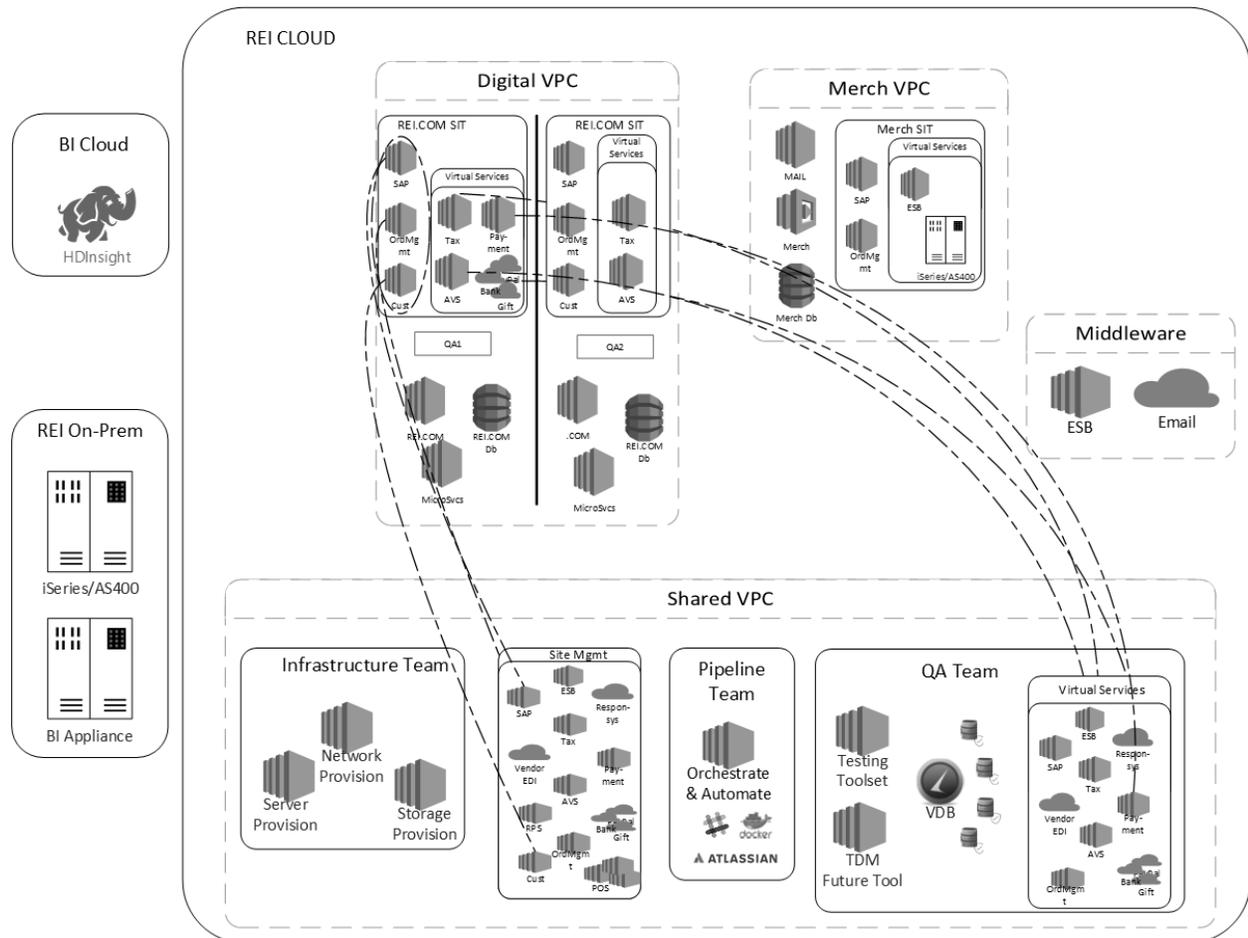


Figure 2

5.2 The Roadmap

In addition to the North Star artifacts, a 3-year roadmap of projects was developed to help keep the focus on reaching tangible milestones in the maturity of the program.

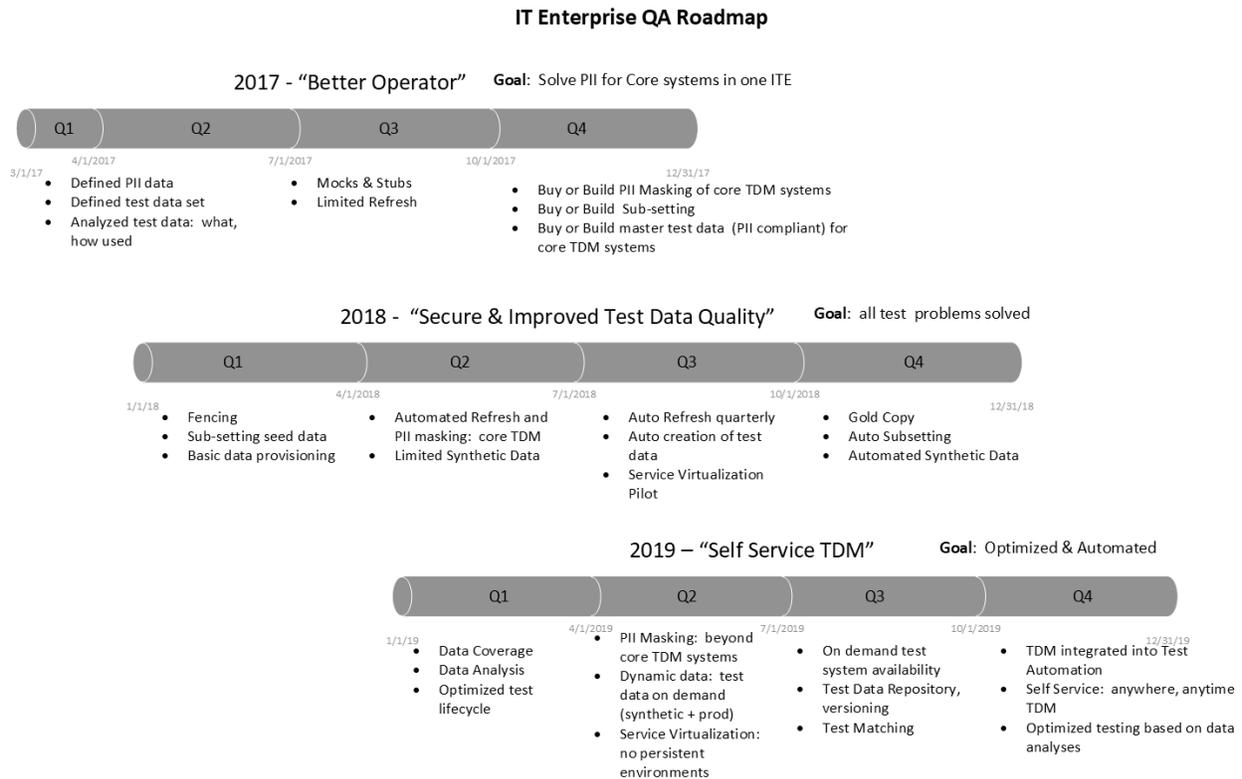


Figure 3

5.3 New tenets brought forward

5.3.1 No persistent environments

In a cloud model there is likely no (or a very rare) use case for having persistent, static test environments. Given the expense of creating most environments, it behooves the organization to know why and when to create a persistent environment, especially if it is on-prem. We understand the usual exceptions are PERF testing environments and, depending on comfort level, prod support environments that mimic the production footprint.

5.3.2 QA does not plan, design or provision infrastructure

QA needs to see themselves as a consumer of infrastructure and yet act as a key partner to help ensure teams get what they need for testing. Regarding the provisioning of whole applications for test, it remains to be seen what role QA will have. The key is that QA will work to provision infrastructure or apps only where there is an ongoing gap.

****Lesson learned #2**** Be ready to redefine what a "test environment" is. Proliferation of servers and databases does not equate to improving your testing capability. Making more copies of systems and environments is usually expensive and unsustainable. Focus on the needed features and functionality.

5.3.3 QA does plan, design and provision platforms, tools and data which enable all forms of testing

This is the “other side of the coin” from point 5.3.2. The focus must be on A) what you can control, B) what enables your core competency, and C) what prepares QA to be a strategic partner for software delivery in the future.

5.3.4 The project roadmap focus

It became the collective QA understanding that the cumulative effect of smaller projects will be significant (something about *sum of the parts*). This shift in thinking was evidenced in the shift away from pursuing the “big bang” (and big dollar) type of projects, and instead focusing on targeted, affordable solutions to use as building blocks

Since early 2017 these artifacts and the evolution in thought have resonated with QA partners across IT, specifically Enterprise Architecture, Site Reliability Engineering, and the software engineer community.

****Lesson learned #3**** Roadmaps are living documents, don't get too attached to them. The combination of the new tenets and re-calibration of the technologies that QA should own resulted in an overhaul of the roadmap after the first year. Present example: we deferred the textbook TDM focus (refresh, subset, synthetic data provisioning) and reprioritized database virtualization and service virtualization projects, because they were achievable with the small team and budget, and were already on the roadmap.

6 Navigating the Projects – Year #1

6.1 Getting going

So the 2017 goal was set: mask the PII data in five systems of record and virtualize the databases for four of those systems (Yes, the math might seem funny, but the DV tool did not support virtualizing an AS400 database, but it can mask it!). The systems chosen to be masked/virtualized were not chosen at random, we focused on the core systems of record that impact the vast majority testing: ecommerce, customer, order management, ERP and warehouse management. There was not a detailed use case review made that drove the decision, it was more of a common sense consensus that drove the decision.

These projects definitely “take a village”. The team chosen for this project included a project manager, a SME/lead, a functional QA analyst, and a QA SDET--plus a substantial amount of support from a senior DBE.

6.2 Value add

The PII data masking depended on the DV platform, so we focus on that first. DV brings with it a lot of goodness that is not present in physical databases, and this is what was socialized as the benefits:

- Access to updated production data
- A regular database refresh cycle can be configured and automated, frequently
- Teams can consume refreshes as needed or choose to retain their changes
- Teams have the ability to bookmark and roll back to previous versions of the database
- Teams can share versions of their database with others
- Vastly smaller storage requirement for each virtual database

6.3 The big challenge—masking

Like any new technology there is a lot to be learned along the way, but the database team had already set the foundation of the DV platform in 2016. Setting up the platform was relatively easy compared to the data masking. To accomplish the database masking, the team followed a pattern for each app:

1. Do the work to get the virtual database ingested
2. Conduct an initial interview with the app teams that knew their own data, and step through a review of the data that appeared to be in the database
3. Develop detailed data rules
4. Execute masking (a very iterative process)
5. Review with the app team to confirm that data masking was successful

6.4 Minor lessons and decisions made regarding our masking project

- While the focus is so often on software developers, when it comes to handling data you will need a ROCK SOLID database engineer (DBE) to help with the database work. There is no substitute for a capable DBE
- The decision was made to retain the vendor's team to accomplish the data masking project. We 'got lucky' and made this decision before we learned the lesson:
 - Your typical database engineer has heard of the various methods and tools needed for data masking in one or more platforms, but you will likely need an experienced data masking specialist to make it an effective use of your time across all the various platforms and structures.
- You will get a big boost from having an analyst (QA functional analyst works great!) that knows the data to help review and accomplish data masking.
- No matter how good you are, databases live on servers and take up space so you will need to be prepared for the cost of infrastructure.
- Project success was defined as successful masking, but standing up an application server and conducting testing was decided to be out of scope for the first project
 - The validation was performed using database query tools and manual inspection, but the decision was made to defer connecting apps to the databases to a later phase
 - This required many iterations of manual inspection, and the number of iterations were increased due to the number of times the VDB's had to be started from scratch.

****Lesson learned #4**** You can't use 'slices' of people's time effectively. Resource availability for our core team was a major constraint, but it did work for each application owner. The SME, Analyst and SDET on the core team were fully allocated to projects, so their time and contributions were constrained. You should plan for a dedicated team to begin masking across the enterprise.

7 Projects Year #2

Rounding the turn into 2018, the team is focusing on several projects: 1) operationalize the DV platform, 2) DV adoption, and 3) SV (Service Virtualization) Pilot.

7.1 Operationalize DV

In a phrase, this is the year that VDB's get real. With PII data masking already largely being accomplished in 2017, the focus for the 2018 Operationalize project is to automate from the point of intake of the production database all the way to the template VDB, from which other VDB's may be provisioned. For this work the vendor was again retained to focus on the tool config required for automation, and scripting required for user security. Again in 2017, our assigned DBE has been critical to project success

7.2 Adoption of DV

The focus of this branch of the project is usage by the application teams. The approach we used is to socialize and evangelize, and we also set a goal of migrating one of the shared end-to-end test environments to VDBs by end of year. We believe the teams will eventually see the need, but we have tried to set some interim goals and ask for their help to achieve them.

****Lesson learned #5**** The benefits of VDB's will sell themselves to some degree, the biggest challenge is getting your champion who will do whatever is necessary to support you supporting them. Our champion from each system saw the value it gives them and was willing to do the work necessary to move their team along the journey with us

7.2.1 DV Successes

Thus far there have been several champions that have stepped forward and gotten their teams using VDBs:

- Marketing/Customer Systems team needing to overhaul annual reward/dividend process - they very happily embraced a masked, stand-alone database for a major code re-factoring effort
- SAP team has "seen the light" and is seeking to replace all non-prod databases with VDB's.

****Lesson learned #6**** Watch for unplanned wins. There is an unintended benefit from the SAP team replacing all their non-production databases: eventually reclaiming ~50TB (\$250,000) of enterprise class storage. The team knew there would be savings, and sometimes you just get lucky on your first try. The flip side of this is be wary of easy wins as there is often some off-setting work required to achieve the win

7.3 Service Virtualization Pilot

This was REI's first foray with an enterprise SV tool. This project is seen as an "enabler" project, which has several major benefits we hope to leverage:

- Enable teams to test without needing the other systems for integrations
- Test reliability, predictability and environment control
- Testing new development before a missing integration dependency is ready for testing
- Elimination of data dependencies
- Strategic value - gets us one step closer to being able to test in the cloud

The approach for this project has been similar to the database virtualization project in terms of it being a collaboration of a key consultant with our core QA team. The difference with SV is that the team is much larger. The first service that was virtualized required four REI resources plus the consultant, researching app config details, going back and forth to a lab of physical hardware, trying various methods to handle security and certificates, etc. It was quite a flurry of activity, but it was a great way to start.

****Lesson learned #7**** - SV will require a broad team with similarly broad skills, knowledge in the various services you want to virtualize, and access to technical configuration information—get as much done ahead of time as you can. Lesson 7b: plan focused working sessions and just make whatever progress you can when you get the right people in a room together.

8 Sharing the Love

8.1 Upping our Social Game

Our team dedicated time in the first half of 2018 to begin spreading the word through REI regarding the benefits of virtualization and QA's readiness to partner with them. The director developed more formal team and personal goals around planned events:

- Quarterly engineering internal meet-ups (good reason to have beer)
- Quarterly "Innovation days": a couple of days of dedicated to team members just trying something and producing something with demonstrated value. "Winners" are eligible to receive some funding to take the concept farther

8.2 Taking ownership

Regarding having vendors lead the way on evangelizing our projects and roadmap, you've heard it said many times, and REI is no exception: "culture eats strategy for breakfast, lunch and dinner"¹. We were engaging the DV vendor to evangelize the products within the company, but we realized that we needed to take the reins and share this with the company ourselves.

****Lesson learned #8**** Our approach of *show, not tell* is proving to be effective at building our most valuable asset: interest and buy-in. You can leverage your vendor partners for a lot but consider making yourselves the face of the projects to your company.



9 Next Steps

9.1 Finishing 2018

As we round the bend into the latter half of 2018, our team is focused on wrapping up the stated project phases, and several other efforts:

- Select one end-to-end test environment and make plan to convert to virtual databases
- Continue the effort to develop a library of virtual services
- Launch an REI EPIC program to recognize and capture savings
- Develop a business case for a formal support program with dedicated resources

¹ https://www.google.com/url?q=https://www.torbenrick.eu/blog/culture/organisational-culture-eats-strategy-for-breakfast-lunch-and-dinner/&sa=D&ust=1532056980167000&usg=AFQjCNEAr5yaEHtgTGjqsJNvikhRJ_s1sA

9.2 The next trip around the sun

In the coming year our group is planning to achieve several expansions to the existing projects. These plans include:

- Design and setup of final SV infrastructure
- Project to finalize full infrastructure for DV platform
- Capacity management and forecasting for DV storage
- Alignment and partnering with the Digital and Site Reliability groups to include virtualization technologies in automated provisioning of systems, apps and pipeline testing
- Self service

10 Summary Conclusions ...

REI's journey is not necessarily unique, but the effect of these streams of work within the company is very encouraging and potentially transformative. In just these few short years QA has gone from being the tail wagged by the dog, to more of a partner and needle-mover within the software engineering ecosystem of the company.

There are a few thoughts to conclude with:

1. Keep in mind this is indeed a journey. What transformed our organization is having a leader who is focused on forward path, and who is willing to accept levels of uncertainty and variability that many 'leaders' won't.
2. Temper the uncertainty by having an overall vision and end state target, and then apply small steps and adjustments, learning as you go.
3. Carve out smaller units of work, be willing to learn from small mistakes, fail fast and fail forward. Make incremental progress and celebrate small wins.
4. You won't have control over a lot of variables, such as key resource availability, conflicts with project and operational priorities (e.g., I can't make your workshop this week because ...fill in the blank). Prepare to just roll with the punches but focus on actually delivering the smaller targets.
5. Expect resistance to some of your ideas and initiatives. As a mitigation, keep the North Star (end state) in focus: don't lose sight of where you're trying to go. Evaluate your progress frequently and be willing to adjust your path—but ensure you are always driving to the North Star
6. Most of all, be positive and confident that you'll get there. And you will.