



Gen and the 5 Rs

Gen and the 5 Rs

Executive Summary

This whitepaper discusses modernisation options related to the 'Gen' legacy technology (formerly CA Gen and, COOL:Gen), a hugely stable work engine that acts as the core of many global organisations, but one that poses a board-level risk due to perceived inflexibility, dwindling expertise, rising costs, and slow development cycles.

Though modernisation strategies can be viewed through the lens of the Gartner "5 R" framework (often described as Rehost, Refactor, Revise, Rebuild, and Replace), there is an industry perception that such Gen systems, due to their scale and complexity, are somehow beyond the reach of such initiatives, as the risk, costs and timelines are assumed to be too high.

TXP, a leader in automated Gen transformation, challenges this view.

With an unparalleled insight into the Gen platform itself, the company uses unique and proven automation to transform applications in-Gen or to

support the organisational journey beyond Gen. Focused Discovery and envisioning exercises look to rapidly understand the current and target states, and TXP's Gen teams are able to support BAU and transitional activity during the transformation programme itself, so easing the impact of change on our end-clients.

Automation is used to re-architect applications to increase strategic optionality - dramatically improving integration options with the Gen systems, isolating functionality to allow functional replacement, and supporting strategic migrations to languages such as Cobol, Java, and C#/.Net.

This paper is aimed at C-level individuals and senior IT decision makers within organisations using Gen and is intended to demonstrate that the "5 Rs" strategies are not only viable for Gen systems, but have been successfully implemented multiple times using proven automation to manage risk, time and cost.



Gen and the 5 Rs

Modernisation strategies around business-critical legacy technology will typically have a complex risk profile. In 2010, Gartner looked to provide a conceptual framework to support the assessment of options. Originally known as the "5 Rs" these were Rehost, Refactor, Revise, Rebuild, and Replace. Though the precise number and naming of these strategies has changed slightly over the years – the key categorisations broadly remain the same.

Gen (now supplied by Broadcom, and previously known as CA Gen, COOL:Gen etc) is a CASE (Computer Aided Software Engineering) tool that was historically adopted by some of the world's largest organisations. Now considered legacy, it still provides a predictable beat at the

heart of many of these organisations globally, while simultaneously being viewed as significant emerging or current risk – due to the common issues of inflexibility, diminishing resource pools, increasing costs and slower development cycles.

We believe a view has developed in the industry that Gen systems, given their scale, complexity and niche-nature, are somehow beyond the reach of the 5 Rs – or at least the risk profile is unpalatable for many of the options.

This short paper looks to explore that view – and in many cases, looks to firmly counter that the individual strategies are not only feasible, but have repeatedly been implemented successfully.

Our credentials

TXP is a global leader in the automated transformation of Gen, with globally unique capabilities and the ability to support any desired Gen strategy.

The first incarnation of our tooling, over 20 years ago, grew out of our ability to unlock the Gen API's, supporting developer productivity and tool-collaboration (such as automated test harness generation and Gen-Microsoft Integrations).

The ability to manipulate the Gen meta-model at scale allowed us to begin to re-architect entire applications, with an initial focus on releasing clients from the limitations of block mode (green screen) systems, automatically transforming them into GUI client/server applications.

This also allowed us to decouple tightly coupled/monolithic applications, creating pragmatic application boundaries based on functional areas, allowing for a range of different strategies for each new independent area.

Although a pro-Gen organisation, client demand led us to expand our tooling to support those organisations that had made the strategic decision to migrate away from Gen, and were looking to develop in native languages such as Cobol, Java & C#/.Net.

Our expertise in tooling automation, coupled with large-scale application development credentials, allowed us to create a solution that delivers hand-maintainable, functionally equivalent code, that is performant and which has no runtime dependencies – the associated tooling, processes and approach are now encapsulated in our industrialised factory model.

Today sees us supporting some of the largest Gen users in the world, successfully transforming their core applications in-Gen and out of Gen, allowing the decades of investment in their core system to be finally unlocked, enabling broader transformation on the application estate.

The 5 Rs in the context of Gen, and TXP's experience of modernising gen application

Rehost (Replatform)



Definition: “Redeploy the application component to other infrastructure (physical virtual or cloud) without modifying its code, features or functions”

By its nature, Gen is able to generate to different target platforms, operating systems, languages, DBMS's, middleware, application styles, etc. This is a huge plus, as it allows different technologies to be exploited as part of a Gen application architecture, it does mean however that, ideally, the application is regenerated using the latest in-support versions of the technologies. TXP can support and advise around this process-to limit the risk (particularly around functional instability or security) of an out of support component.

A replatform exercise may have the objective of simply moving the application 'off-mainframe and onto a distribute environment, to support cost-saving targets, or increase flexibility. More commonly, a replatform may also be part of larger ambitions to transition the application workload to the Cloud. Though eminently feasible, challenges must be considered around many aspects of a replatform:

- Application architecture aspects may be suited to one environment and not another
- Re-factoring as part of a replatform
- Use of proprietary elements of the current environment
- Move to a new language and the need to rewrite

- Communications
- Different security solutions
- Batch framework and scheduler choice
- Ability to cope with volumes
- Non-functional requirements, for example, performance and security
- DBMS choice and potential data migration
- Approach - phased or big bang
- Timeframe for modernisation
- Application/environmental coexistence
- 'As is' resource skill set and alignment with 'To Be' solution
- Availability of supporting app dev tooling to support the likes of a DevOps approach
- Quality assurance to demonstrate functional equivalence following replatform

A strategy covering all of the aspects listed and more is required to ensure a complete view of any proposed replatform is understood to ensure the greatest possible chance of a successful programme outcome.

TXP has supported a number of replatforming exercises globally, and this is always underpinned by a short discovery exercise (4-6 weeks typically, depending on scale). This, along with our proven processes and tooling, provides for much greater certainty to what is historically an area that is perceived to be laden with risk.

Definition: We re-envision the Gartner definition of “Restructure and optimize existing code (although not its external behaviour) to remove technical debt and improve nonfunctional attributes” to be a broader definition of “using the Gen meta-model to regenerate the entire Gen application into hand-maintainable native code, that can then be deployed to the Cloud, on-premise or hybrid infrastructure of choice.”

For those organisations looking to convert all or part of their Gen estate into native hand-maintainable source code (often Java, Cobol or .Net) this option can provide a scalable solution to ultimately move away from the dependence on the Gen platform as a whole.

Key aspects of such a strategy include maintainability, functional equivalence (we target 100%) and speed of conversion.

Maintainability

Conversion through configurable automation is a must, to maintain high levels of quality at scale - ensuring that the generated code is aligned to client standards, and so meaningful to the client's development teams.

A deep Gen specialism is also necessary in any transformation partner - as often it is necessary to prepare the Gen estate prior to

conversion - removing redundant models, resolving model corruptions and potentially simplifying the Gen architecture to ease future change and release efforts.

Functional equivalence

This can only be achieved when the conversion is based on the Gen meta-model, and not by using Gen's own generated code to seed transformation. This is because Gen code will naturally call proprietary Gen runtimes, which would not be available in any converted code base. We have created native equivalents of all Gen runtime capabilities - and develop equivalents when we uncover client-specifics.

"This is supported by a process and tooling that has been exercised, evolved and improved over 20 years"

Speed of conversion

This is typically important to support the transformation ROI - reducing the distraction of the transformation on key client resources. This is supported by a process and tooling that has been exercised, evolved and improved over 20 years. In that time it has created many million lines of code that are now supporting organisations over 5 continents. Though tooling dramatically accelerates such programmes, standard processes allow ongoing business change to be ingested into the transformation - so allowing the business to remain competitive and relevant in the market.



Definition: “Restructure and optimize the existing code (although not its external behaviour) to remove technical debt and improve nonfunctional attributes.”

This is the perfect example of where many organisations have historically built up a view that their Gen application is beyond the reach of the 5 Rs. Many Gen users have large numbers of tightly coupled applications built around a central shared data model/database. This was common development practice in the 80s and 90s, however, this architecture, with its high level of coupling and interdependency, now presents these organisations with a very real issue. Both flexibility, and cost/speed of change are typically impacted, to a level that some believe is almost impossible to overcome.

TXP has a deep understanding of Component Based Development (CBD) - and created globally unique automation over a decade ago to support organisations who were moving towards it. Over time however, CBD has become viewed as more of a purist methodology, with TXP responding accordingly, applying our pragmatism to create what we call 'Functional Area Isolation' (FAI).

"it is a more measured, fit-for-purpose use of those principles, futureproofing your application estate"

Although based upon sound CBD principles, it is a more measured, fit-for-purpose use of those principles, futureproofing your application estate by separating functional areas in a way that makes sense to the business. The alignment can be based on identified business- related boundaries or on more technical boundaries. The decision depends on the rationale for the re-architecture and the desired outcomes and benefits.

The intention is for it to be a pragmatic process that works for the specific needs of a customer rather than a generic standard that might contain either compromises or too much complexity for a particular customer.



During 'Discovery' the basis for isolation would normally be agreed with a customer. The impact of this on a Gen estate would be modelled so that the impact (and benefits) could be assessed. Once agreed the associated approach, plan and timeframe would be the natural outcomes of Discovery.

By rearchitecting the Gen system into business-aligned components, 5R aligned strategies can be implemented per component - which may include replacement with a COTS package, selective migration/rewrite, or 'leave as is'. By far one of the most critical benefits of FAI - that effectively unlocks key 5R options, is the ability to transform in a phased manner, without the need for a high-risk big-bang implementation - something that the vast majority of organisations look to avoid if at all possible.

It is also worth noting that the automated re-architecture of such systems can be applied not only in support of the creation of business-aligned components, but also in the technical re-architecture of historically tightly coupled areas of the system. With Block Mode applications for example, the presentation, business logic and data access elements are all tightly coupled. Re-architecture of such applications to a more flexible architecture

provides organisations with a greater freedom of choice when considering modernisation options. TXP's automated solution re-architects Block Model applications into GUI, Client/Server applications within Gen. This opens the modernisation options up, facilitating the replacement of the UI with something more modern, exposure of the business logic as web services for consumption outside of Gen, etc.

Rebuild (Rewrite)

Definition: “Redesign or rewrite the application component from scratch while preserving its scope and specifications”

Given that Gen systems often encapsulate decades of client-specific business logic and data, and sit at the heart of the organisation - a rewrite is often seen as a 'last resort' - but is undertaken on occasion.

This is most relevant where organisations are looking to transform their operational processes, so are removing themselves from the reliance on old processes and associated systems.

Consideration should be given to core business rules, workflows and algorithms, that may sit at the heart of the Gen application, which would be disproportionately difficult to rebuild, and ideally should be surgically extracted for reuse in the new application.

Proven analysis techniques combined with automation can allow these capabilities to be isolated and extracted, representing them in native code for inclusion in the rewritten system - reducing the cost and risk of rebuilding what is effectively the client's historic intellectual property.

TXP has supported many clients that were pursuing a manual rewrite. Specifically, we have tooling around automated model analysis, functional isolation and architectural analysis, identification of redundant code or corruptions (that can subsequently be removed via automation) that will act as accelerators. We can also assist with the migration of Action Diagrams / Operations / Transactions to use a back-end processing in any new system. All of these are ways to accelerate and de-risk this approach.





Replace (Retire)

Definition: “Eliminate the former application component altogether and replace it, considering new requirements and needs at the same time”

By combining some of the modernisation capabilities listed above, it is possible to manage the risk of retiring all or some of a legacy Gen application.

Re-architecture of Block Mode Applications

This approach enables choices to be made about the UI and the consumption of the business logic. New UI solutions can be introduced that can still exploit the existing business logic (Gen servers) which can be exposed as web services. This enables organisations to continue to exploit the investment made in their business logic over many years. It is possible to deploy elements of an application to different environments given the more flexible architecture.

Functional Area Isolation

This approach enables organisations to structure their Gen application estate in a way that facilitates their modernisation goals whether that be refactoring, rewriting or replacement. Once an application has been isolated along agreed boundaries, different decisions for modernisation can be made for different areas.

The implementation of standard interfaces means communication with any area is standardised. Encapsulation means areas can be gradually retired/replaced with minimal impact on the wider estate.

The flexibility offered by the processes of application re-architecture and/or Functional Isolation means that an organisation is able to take control of the modernisation journey.

Facilitating your 5R Gen Strategy

When bringing any 5R strategy to life in the Gen space, there are three supporting pillars that should be considered.

Technical strategy

Having supported the 5R outcomes for over two decades, we are able to advise on the opportunities and risks associated with each potential direction. Proprietary tooling provides a detailed cross-portfolio view of your Gen estate, to allow for a confidence evaluation of the current situation.

This then acts as the base for our team to work with you to identify the most applicable options of the 'to be' state- considering your key drivers around cost, time and risk. Typically, we then collaborate with your key stakeholders to construct the technical strategy, that then informs the subsequent transformation.

Transformation strategy

Gen systems are typically business critical, touching many other areas of the application estate. The technical solution must be considered as being part of a wider business solution - often involving changes in the delivery of ongoing change (for the better).

We typically work with your heads of transformation to advise on how best to manage change in the Gen area- including phased approaches and how best to structure your change pipelines.

Resource strategy

Though often seen as a subset of a Transformation strategy, resourcing is so critical to the success of a transformation, that it is called out separately here. Typically, your Gen specialists will need to engage with any transformation, so distracting them from scheduled Change activity. We can support 'BAU' change, while also embedding our Gen specialists into your teams to help steer the transformation activity - typically by successfully combining onsite and remote working.

The resource strategy can include the transition of your Gen resources, with their years of domain experience, into your new architectural target.

Conclusion

Though the scale and complexity of Gen systems may make any 5R strategy seem like a pipe-dream, our combination of tooling, expertise and people makes Gen transformation not only feasible, but actually a realistic opportunity to leverage what is often a historic multi-decade investment on the part of our clients.

A rapid discovery exercise will typically inform options, timelines and costs- allowing you to prepare for what can be an exciting phase of your organisation's evolution.



Technology
X People

TXP has worked with clients on five continents to support and modernise their Gen estates, rearchitecting block-mode applications and isolating functional areas to support digital enablement, making future business services scalable and flexible.

Clients wanting to migrate to native technologies benefit from our proven automation approach, with training and consultancy completing our holistic service.

Automated modernisation of a number of large Gen applications to convert them from mainframe block mode to web client/server. Involved splitting block mode procedures and generating new UI clients to access new mainframe servers in support of a web enablement project.

Use of automation to perform Gap Analysis in support of COTS package adoption, alongside functional isolation in the realignment of functional boundaries.

Responsible for all Gen support (2nd/3rd/On-Call) and new development/enhancements of a mature Gen user via the provision of onsite and remote support.

Gen conversion to functionally equivalent, hand maintainable native code.

Large functional isolation exercise to rearchitect a significant monolithic and tightly coupled application into a number of smaller component models.

TXP welcomes the opportunity to engage with Gen users worldwide, to help them understand how Functional Isolation can be implemented, allowing them to reduce risk, increase flexibility and ensure that they are ready to deal with any eventuality.