CGI RCDA:

A responsive, collaborative approach to digital architecture

CGI

Contents

Transforming into a digital organization involves more than the deployment of new digital systems. It also requires changing the way an organization acquires, delivers and operates its systems.

In the past, all of the processes involved in managing systems operated via top-down, gate-controlled governance. Today, they increasingly rely on collaboration rather than control, and, as a result, have become more responsive to business and technology changes.

This fundamental change in systems management is necessary to achieve business agility, and it has had a profound impact on the digital architecture discipline. RCDA is CGI's digital architecture approach. It aligns closely with this new way of working, transforming the architecture function of organizations worldwide.

The need for a more adaptive architecture

Back in the days when markets and economies were more stable, and changes came less fast and frequently, organizations could predict and plan their operations with reasonable accuracy. Up-front planning and architectural design were the norm for building a stable foundation for growth.

Nowadays, change is less predictable and more frequent. We live in a world where organizations need to continuously sense what is going on and promptly respond. As a result, they also need a different approach to architecture—one that is responsive to a world characterized by Volatility, Uncertainty, Complexity, and Ambiguity (a VUCA world). They also require an architectural foundation that is not a slab of concrete designed up-front, but a continuously adaptive landing zone for innovation and new business features.

As technology enables organizations to disrupt, it also is creating a more dynamic and competitive marketplace. This requires organizations to change continuously at speed, become more agile, and innovate. The 2020 CGI Client Global Insights confirms that organizations recognize the need for agile transformation, but many also report an increase in cultural and organizational resistance (83%), legacy and agility challenges (70%), and a lack of funding to transform (43%). Most organizations cite a substantial gap between strategy and agile delivery. While 91% of organizations have a digital strategy in place, only 12% are producing results at the enterprise level.

The 2020 CGI Client Global Insights confirms that organizations recognize the need for agile transformation, but many also report

an increase in cultural and organizational resistance

83%

legacy and agility challenges

70%

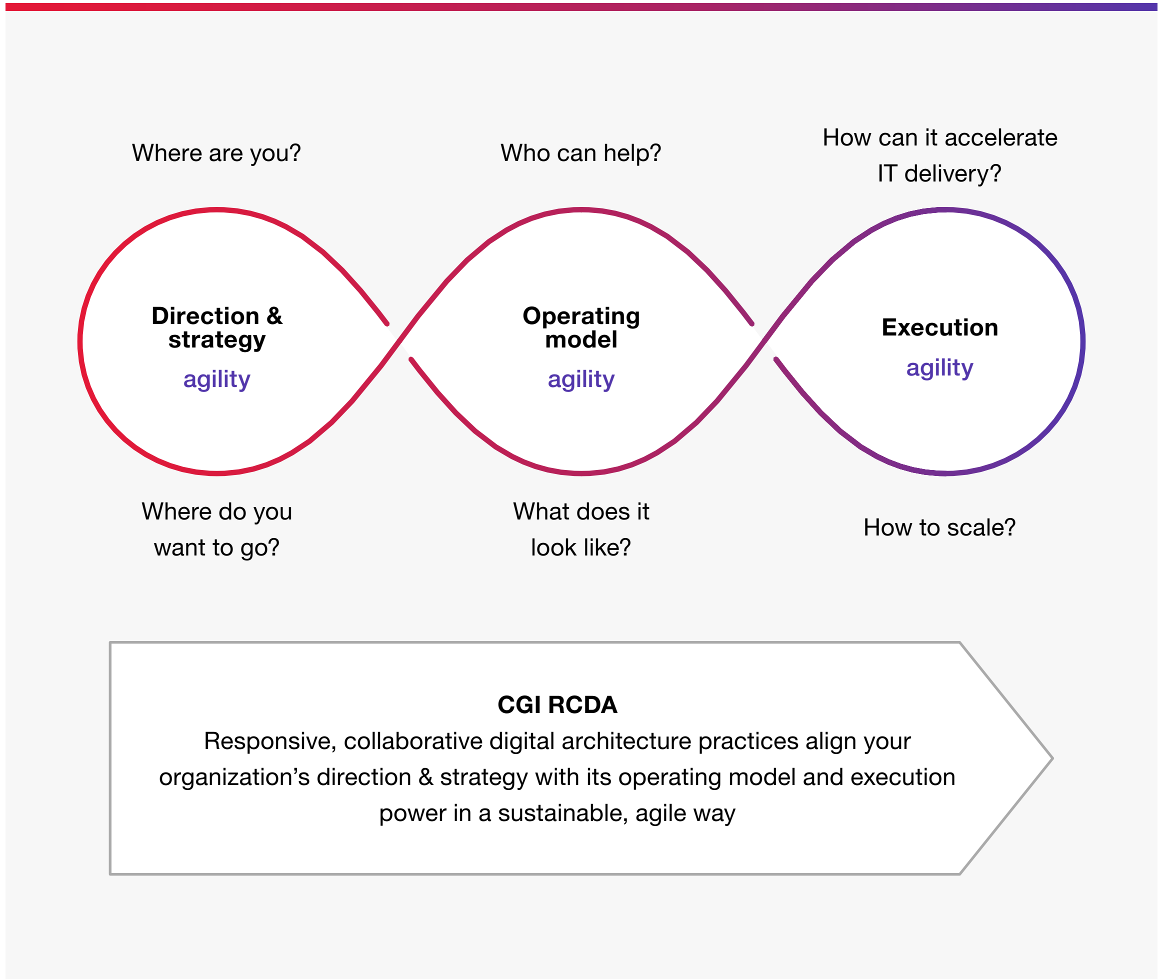
a lack of funding to transform

43%

Business agility and architecture

Over the last five years, we have been researching business agility and its impact on value creation. We have spent time with some of the world's leading organizations to understand their experiences and lessons learned with respect to business agility. Our research highlights the importance of balancing collaboration and autonomy in design practices. Both are key principles that CGI's own architects have adopted to be more responsive and collaborative in an agile context. In this paper, we explore this agile way of working and introduce CGI's agile architecture approach—Responsive, Collaborative Digital Architecture (RCDA).

Business agility and architecture often seem to be opposing forces. A heavy, up-front design process moves too slowly to cope with the speed of change in the world, and architects often have a reputation of being disconnected from the reality of agile teams.



CGI RCDA

Responsive, collaborative digital architecture practices align your organization's direction & strategy with its operating model and execution power in a sustainable, agile way

The lack of architectural coherence in a digital landscape can severely impact agility, as well as budget.

On the other hand, as many organizations have experienced, the lack of architectural coherence in a digital landscape can severely impact agility, as well as budget. We have seen a significant shift in attitudes toward the architect's role. Many organizations now prefer to allocate the responsibility for major design decisions to teams rather than to a named architect, who takes on the role of *pathfinder*, *master builder*, *ninja developer* or *steward*. In such cases, the *role* of an architect may have disappeared, but the architecture function still exists; it exists as a set of collaborative responsibilities allocated to other roles or teams.

Successful architecture in a digital world

Effective design is key to a successful digital strategy; a strategy in which design decisions are not made up-front by a centralized architecture board, but by a team that continuously learns from new insights generated in an ever-evolving IT landscape.

Our experience with successful agile architecture teams has led to the defining of five responsibilities that lie at the heart of RCDA.

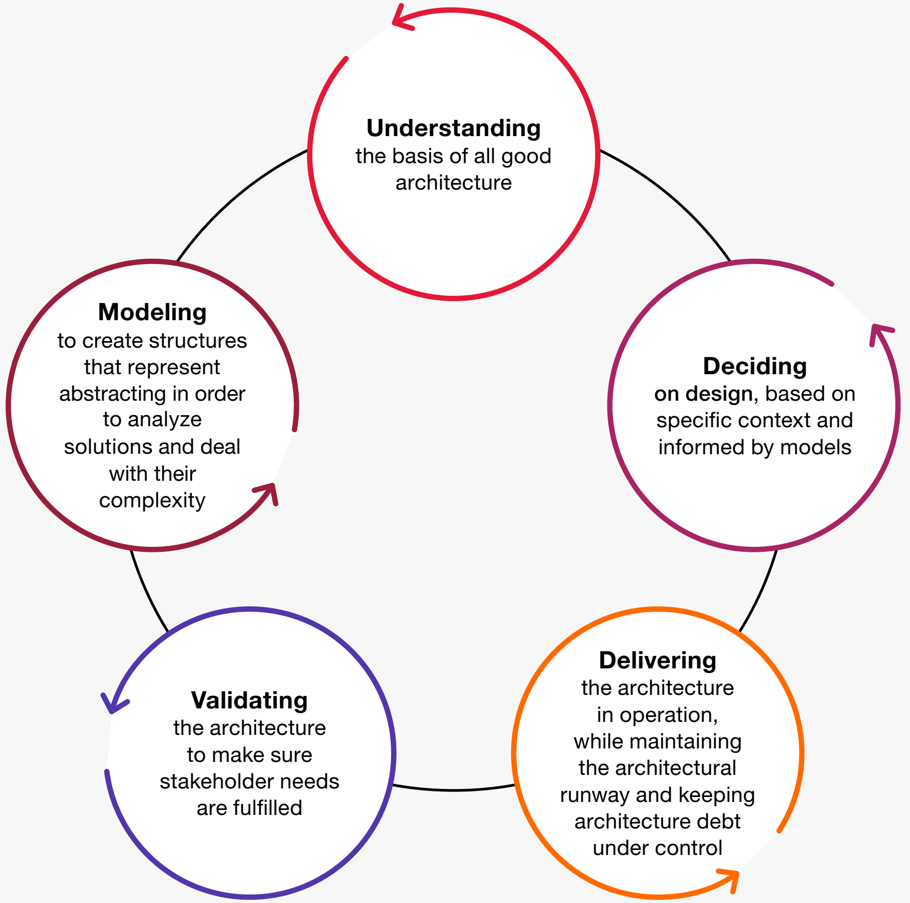
Architecture responsibilities

Research [1] shows that applying architecture practices significantly improves the quality of software solutions, along with the risk and cost control of their delivery. If organizations want to reap these benefits without having a named architect, they need to consider the maturity of the architecture function on an organizational level. In RCDA, we see this function as a set of responsibilities: understanding, modeling, deciding, validating, and delivering. This model helps organizations assess how well they are doing in terms of their architecture and where they can improve—with or without named architects. We fine-tuned this maturity model by applying it in practice for years, and it has proven to help organizations recognize their weak spots and find ways to improve.

Fulfilling the five responsibilities in isolation is not enough: they should be fulfilled in a coherent way.

- **Understanding** context, the basis of all good architecture
- **Deciding on design**, based on the specific context and informed by models
- **Delivering** the architecture in operation, while maintaining the architectural runway and keeping architecture debt under control
- **Validating** the architecture to make sure stakeholder needs are fulfilled
- **Modeling** to create structures that represent abstractions in order to analyze solutions and deal with their complexity

The five responsibilities of the architecture function


Understanding
the basis of all good architecture
Deciding
on design, based on specific context and informed by models
Delivering
the architecture in operation, while maintaining the architectural runway and keeping architecture debt under control
Validating
the architecture to make sure stakeholder needs are fulfilled
Modeling
to create structures that represent abstracting in order to analyze solutions and deal with their complexity

Five responsibilities: background

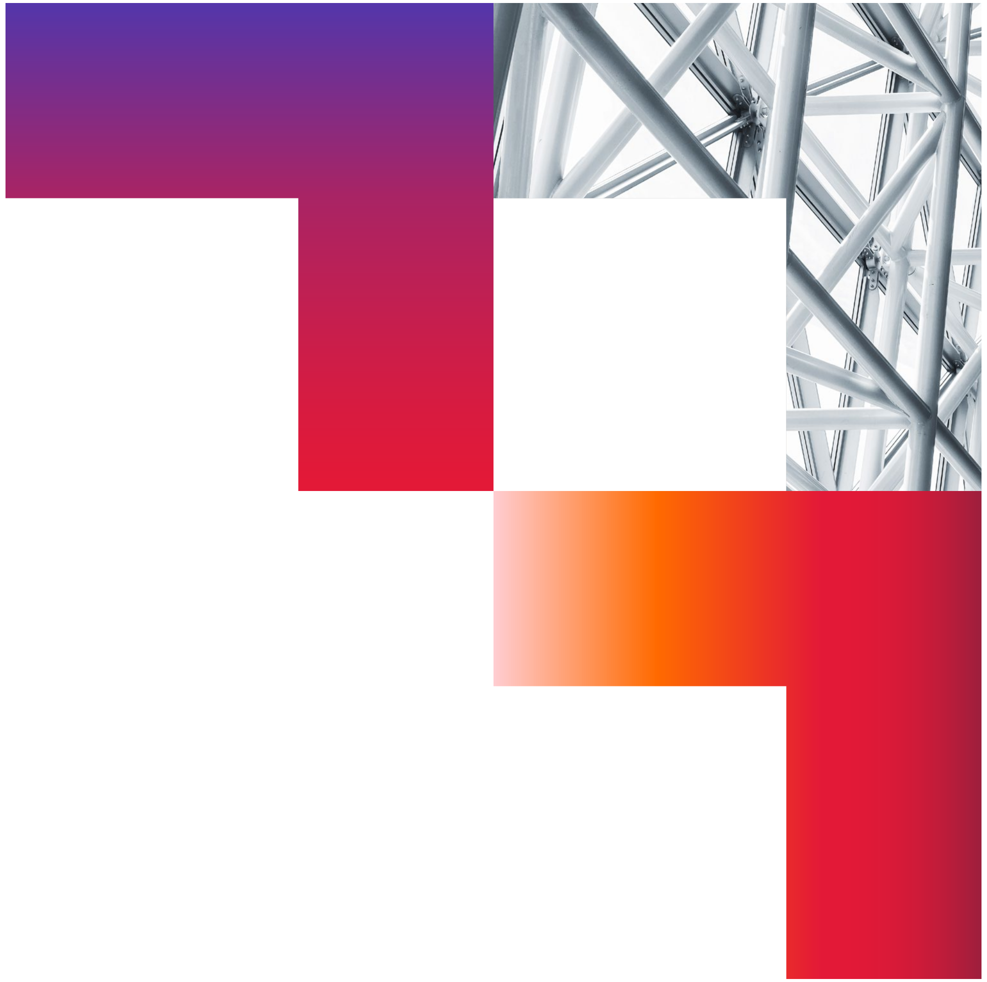
In the 1990s [2] architecture was viewed as a set of structures that represent an abstraction of a system being delivered – an abstraction needed to deal with the growing complexity of typical software systems. The main architectural activities were [3]:

1. **architectural analysis, with the aim of *understanding context*;**
2. **architectural synthesis, resulting in architecture *models*;**
3. **architectural evaluation, aimed at *validating* the architecture**

In the early 2000s, a second perception emerged, with a new responsibility focus: architects needed to make important *decisions* [4] in order to create the right models of their solutions. If *structures* describe *what* the architect creates, the decision-making refers to *how* they create it.

Around 2010, a third perception emerged: the *why* was added to the *what* and the *how* of architecture. This view shed light on the business goal of architecture: to improve organizations' control over risk and cost [5] – not only during design, but extending the architects' responsibility to the *delivery* domain.

So we end up with five architectural responsibilities: understanding context, making decisions, modeling, validating and delivery. Fulfilling the five responsibilities in isolation is not enough: they should be fulfilled in a coherent way.



Balanced architecture

The best architectures result from paying proper attention to all five responsibilities described above. This is not easy due to factors like cultural pressures, dogmas and misconceptions. Many organizations ignore some of the responsibilities, resulting in a flawed architecture function. Two extreme examples are the Waterfall Wasteland and the Agile Outback caricatures described below.

Paying *proper* attention to all five responsibilities, however, does not mean always paying *equal* attention. Depending on the context, modeling may indeed require more attention than decision-making, and validation may be more critical in some situations than in others.

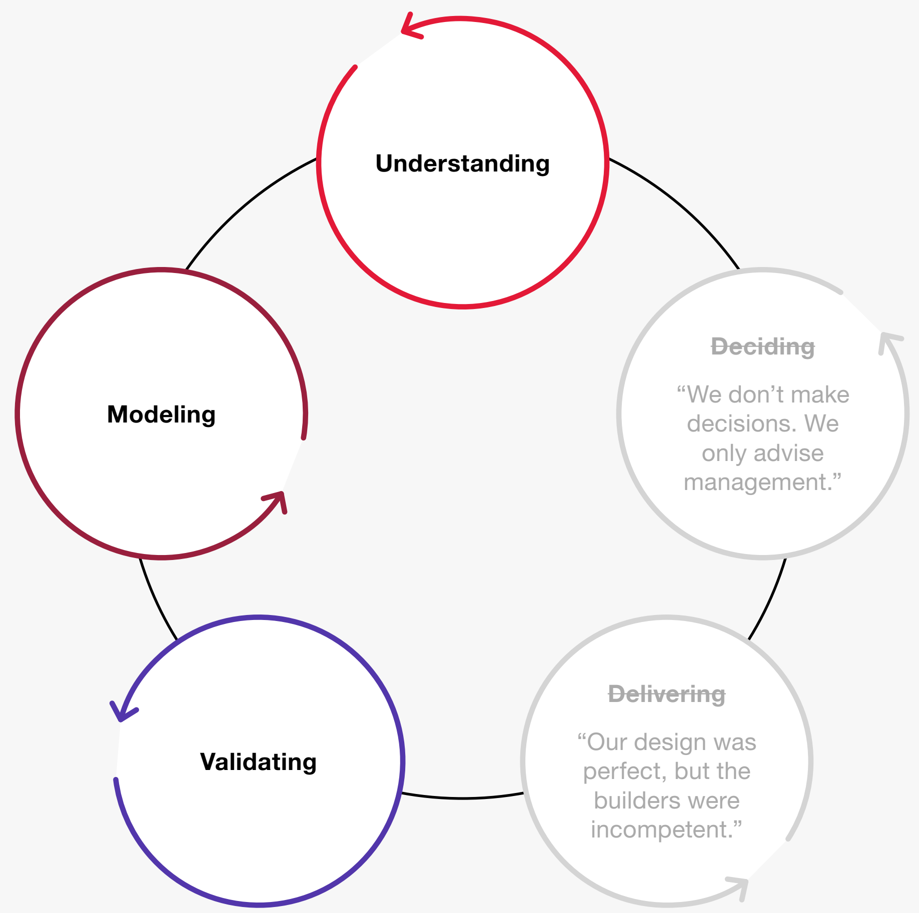
When talking to teams, architects, and stakeholders in different organizations, we noticed some interesting patterns in the way they took up these responsibilities. We created caricatures to highlight the differences among those patterns and called them the *Waterfall Wasteland* and the *Agile Outback*. Please note that they are caricatures; they do not exist in real life, have exaggerated features, may be amusing to some and offending to others, but they can be useful in making a point.

Caricature one: Waterfall Wasteland

In the Waterfall Wasteland, architects sometimes live in an ivory tower. They ignore decision-making and delivery responsibilities, considering them to be someone else's problem. They have a very clear job description—to create perfect models and validate them against stakeholder needs. If the resulting solution is unsuccessful, it's obviously not their fault. The idea that they would be responsible for decisions or share responsibility for successful delivery is abhorrent to them. It would mean that their success would depend on the capabilities of others.

Organizations in the Waterfall Wasteland typically have trouble adapting to change. The carefully modeled and validated designs have a limited shelf life and are hard to adapt to new insights gained during delivery. There is a long feedback cycle between architecture and delivery. The (often hefty) architecture documents go out of sync with reality, and become ballast and waste.

Figure 2 Waterfall Wasteland

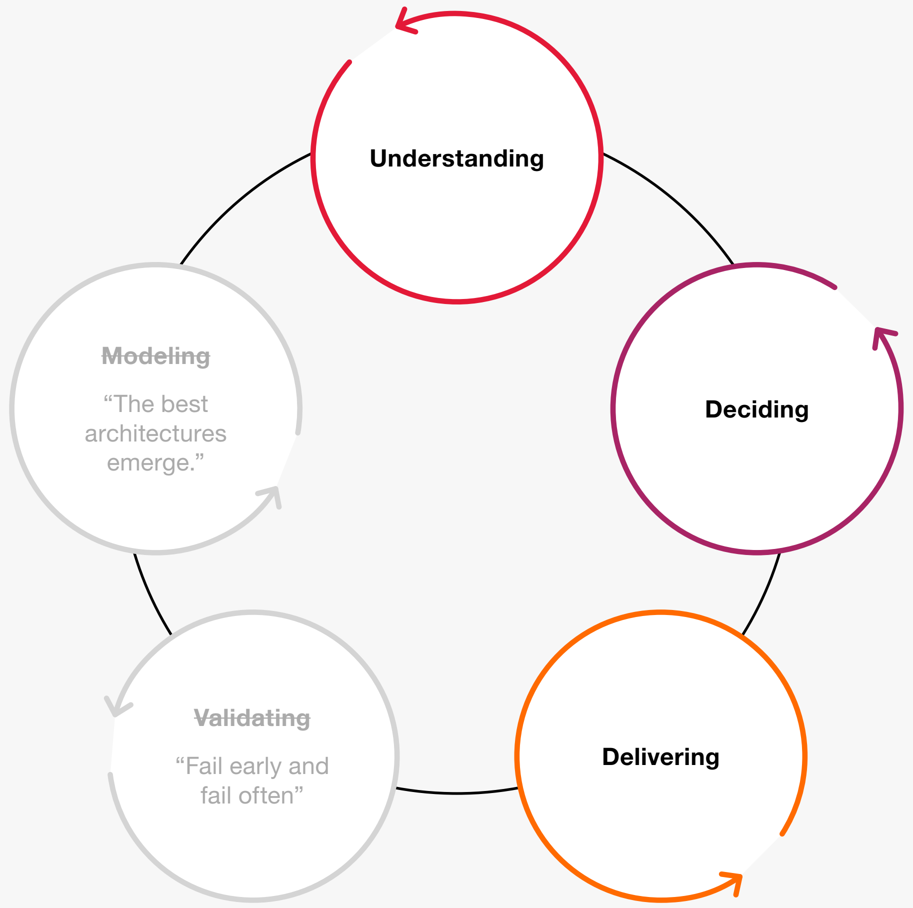

Understanding
~~Deciding~~
"We don't make decisions. We only advise management."
~~Delivering~~
"Our design was perfect, but the builders were incompetent."
Validating
Modeling

Caricature two: Agile Outback

In the Agile Outback, teams usually do not have architects. They avoid modeling since, according to the Agile Manifesto,[1] “The best architectures… emerge from self-organizing teams.” This could be misinterpreted to mean that modeling is unnecessary or even counterproductive. Teams in the Agile Outback rarely reason about or validate designs using models. Instead, they rely on quick feedback from failures.

Organizations in the Agile Outback produce a lot of direct business value at high velocity in the beginning of a product’s life cycle, but in our experience, they tend to have problems sustaining that velocity. They often have to revisit decisions and rework due to the lack of forethought. Some architectural decisions are not easy to refactor, and a few hours spent generating and evaluating alternatives are well worth it.

Figure 3 The Agile Outback



[1] http://agilemanifesto.org

Successful agile architecture

How can organizations avoid the Agile Outback or the Waterfall Wasteland? How can teams find the right balance? How can they work within the *Goldilocks Zone* with an adaptive architecture?

Over the years, CGI's architects have developed and extensively validated four key principles that help organizations become effective at agile architecture:

1 **Shorten your architectural feedback loop**

2 **Focus on business impact**

3 **Achieve "just enough" anticipation**

4 **Architect as a team**

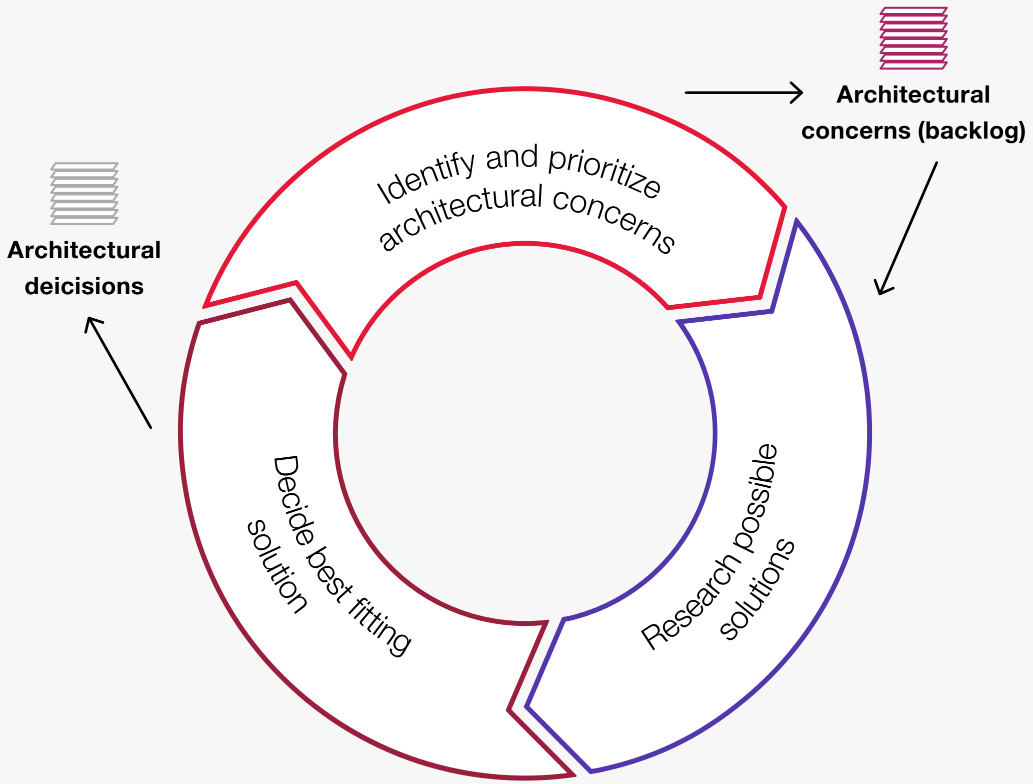
Shorten your architectural feedback loop

Perhaps the most vital lesson architects can learn from the agile mindset is the importance of short feedback loops. The quicker we receive feedback on an architecture, the faster we learn about its effect within a specific solution context. Architecture is a matter of reducing uncertainty by gathering knowledge and making decisions, and a shorter architecture feedback loop speeds up that uncertainty reduction, leading to better architectures. On top of this, shorter loops lead to shorter reaction times when things change, which increases business agility.

An agile architecture is not a "big up-front design," but rather a continuous stream of architectural decisions, made step by step.

The secret of a short architectural feedback loop is to change our view of the main deliverable of the work. An agile architecture is not a "big up-front design," but rather a continuous stream of architectural decisions, made step by step. This helps to control the uncertainties and risks surrounding complex digital solutions. How much architecture to build is determined not by agile dogmas like, "You Ain't Gonna Need It" (YAGNI), but by economic trade-offs, taking into account the real value of architecture in context. The key change we need to make is to no longer view architecture as a design document for projects, but as a continuous decision-making process for gaining control over costs, risks and uncertainties with a short feedback loop. Only then can architecture deliver the value add and flexibility required by the modern digital world.

RCDA's Architecture Microcycle: a backlog of architectural concerns increases agility


Architectural concerns (backlog)
Identify and prioritize architectural concerns
Research possible solutions
Decide best fitting solution
Architectural deicisions

Focus on business impact

A key benefit of using a short feedback cycle in handling a backlog of architectural concerns is that we can quickly reprioritize the architecture work when circumstances change. Most of our attention should focus on concerns that have the highest business impact. This impact can consist of enabling new business value and opportunities, but very often it is about risk and cost control. This is why RCDA originally stood for Risk and Cost Driven Architecture.

Achieve "just enough" anticipation

How do we determine the right amount of architecture? According to the first principle above, architecture is a flow of architectural decisions made as part of a short feedback loop. This flow should be ahead of solution development and delivery with "just enough" anticipation.

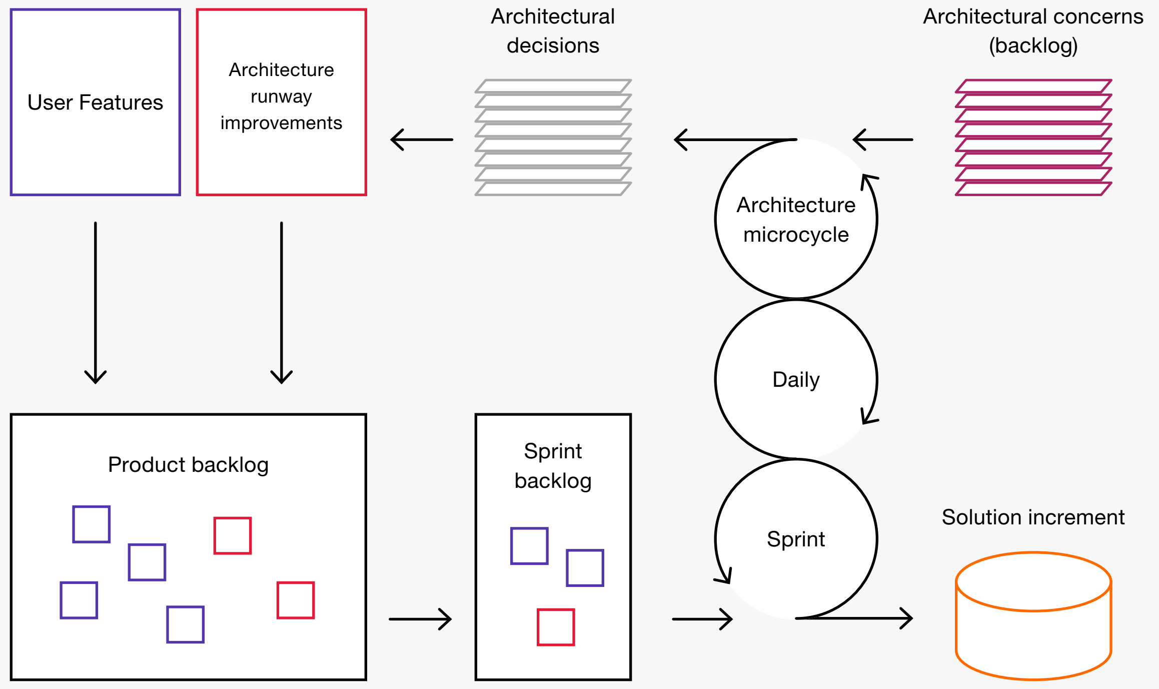
The Scaled Agile Framework® uses the metaphor of a runway that is continuously being extended while in operation, so that it is always just long enough to accommodate the new planes that are anticipated (the planes in the metaphor are upcoming solution requirements). The new, bigger planes can land only after the runway's extension. Dependency analysis determines which runway extensions are required to land which planes.

Sometimes you may temporarily extend the runway with an inferior material for the sake of speed. This represents technical debt that you will need to repay (repave) at some point to prevent accidents. You should base all decisions (when to extend or repave the runway) on sound economic reasoning.

Architect as a team

Depending on their appetite for central coordination (often driven by the complexity of their digital needs), digital leaders may have dedicated architect roles or no architects at all. Organizations apply RCDA practices by embedding the principles of agile architecture into their ways of working, irrespective of whether they have "architects, “architecture owners” on teams, or "crowdsourced" architectural decisions. A key consideration is that the consequences of architectural decisions affect the delivery (agile or DevOps) teams, not as commands from a higher authority, but as user and enabler stories that extend the architecture runway with “just enough” anticipation.

Aligning the Architecture Microcycle with the scrum cycle to facilitate collective architectural decision-making



Your RCDA journey

Every organization is different, and the steps for improving your architecture will be specific to your organizational context, history and goals.

Here is a view of what your RCDA journey could look like.

Orientation

One of CGI's experts presents RCDA's principles and practices to your architecture leadership. We discuss your challenges, whether they lie in the Waterfall Wasteland or closer to the Agile Outback, and start plotting a possible path to more fertile grounds. We select one or two domains to start the journey. In a 90-minute meeting, we inform key architecture players, including business and delivery stakeholders, about a new vision for responsive and collaborative design and share an outline of the journey to get there.

Training

We immerse key players in the architectural design and delivery process in RCDA material during a three-day interactive practitioner course filled with knowledge, examples and exercises.

RCDA maturity radar

It is crucial for organizations and teams to know where they are in terms of agile architecture maturity. RCDA comes with a maturity model [2] to assess how teams and departments are fulfilling the five responsibilities. This assessment helps to identify weak spots in their architecture function, but also serves to measure progress on their journey towards higher maturity. The resulting scores reveal where the team is already strong, and in which areas change can generate more value.

The architecture leadership, facilitated by CGI's experts, compares the maturity baseline with their own ambitions. Together, we create a backlog of concrete activities to close the gap between reality and ambition, prioritized by business value.

RCDA and coaching

Once we identify areas for improvement, we can use RCDA's principles (as outlined above) and practices as guidelines for growth. A transition to agile architecture moves part of the architectural mandate from a central governing team to local delivery teams. Implementing tooling such as central and local architectural decision registers and knowledge repositories facilitates this step. Guidance on combining evolutionary architecture with just enough risk and cost governance also supports the transformation.

Every few months, teams that have embarked on the journey to agile architecture maturity get together to share good practices, identify obstacles and continue to the next iteration. More domains join the journey, led by agile architecture champions appointed in your organization.

After the initial transformation (typically 3-12 months, depending on the organization's size), new maturity assessments show the progress that has been made. Additional improvement activities as indicated by the assessments led by your own agile architecture champions. From time to time, we will share new insights gathered from RCDA users globally with your architecture leadership, leading to continuous improvement.

Your RCDA journey can stand on its own, but also can be part of a wider business agility transformation

CGI Business Agility Maturity Assessment

CGI's Business Agility Maturity Assessment helps today's complex enterprises operating in dynamic environment. Using our detailed 190 capability statements, we assess an organization's current state across 3 dimensions:

1. **Direction and strategy**
2. **Operating model**
3. **Agile execution**

To ensure enterprise-wide alignment, agility, and resilience, an organization must manage the "flow" across these dimensions. The 14 identified golden threads are cross-dimensional themes, which will be assessed within 9 fish-bowl events or a digital survey.

Benefits

RCDA is an approach that has built an impressive track record in a relatively short time, helping large organizations modernize their architectures. Its benefits, extensively researched and published [3], include:

- A modern view of architectural design that complements the speed and flexibility of agile development and helps organizations find the balance between long-term predictability and quick business value by enabling “just enough anticipation."
- Support for teams in gaining control of risks and finding a “sustainable pace” that prevents excessive build-up of technical debt.
- Creation of an environment where architects base design choices on a clear and agreed upon understanding of the business context, using objective and economically oriented trade-offs, rather than hypes or personal preferences. RCDA stimulates such an environment by introducing practices that objectify architectural decisions and priorities and put them in a business context.

- Enhancement of the quality of solutions. RCDA practices contain guidance for early and effective evaluation of a solution's quality attributes and other key requirements.

- Transparency in solution costing structures. RCDA provides traceability from architectural requirements to the costing model.

RCDA contains 13 practices to improve architecture effectiveness. These practices received international recognition when the Carnegie Mellon University's Software Engineering Institute conferred its prestigious Linda Northrop Award to CGI thought leader Eltjo Poort for his work on RCDA in 2016.

1,500

Architects trained

13

Proven practices

10

years of knowledge and experience

14

Peer-reviewed publications

Open group

Recognized in the certified architect program

Linda Northrop Award

Software Engineering Institute

Conclusion

Architecture is all about design decisions that have the highest impact on digital solutions. RCDA is an architecture approach developed to close the gap between architecture and the agile mindset. It combines the extensive scope of enterprise architecture with the pragmatism and agility of modern software development methods.

RCDA offers especially effective guidance in translating architectural concerns and priorities into business terms like cost, risk and value, enabling architects and teams to communicate more effectively with business stakeholders. RCDA practices based on a powerful set of agile principles and accompanied by extensive guidance on how to apply the approach in various frequently occurring contexts.

To learn more, visit cgi.com, or contact info@cgi.com.

Architecture in the digital world is an essential discipline for safeguarding the quality and sustainability of modern, complex digital solutions. Architecture does not need to obstruct agility. RCDA offers a proven architecture approach that is well suited to today’s agile business needs. In addition, CGI offers RCDA consultancy, training and tooling.

References

[1] R. Slot, “A method for valuing architecture-based business transformation and measuring the value of solutions architecture” (PhD Thesis), Amsterdam: University of Amsterdam, 2010.

[2] E. R. Poort, “Between the Waterfall Wasteland and the Agile Outback,” *IEEE Software*, pp. 92-97, Jan-Feb 2020.

[3] E. R. Poort en H. van Vliet, “RCDA: Architecting as a Risk- and Cost Management Discipline,” *Journal of Systems and Software*, pp. 1995-2013, 2012.

[4] M. Shaw en D. Garlan, “Software Architecture - Perspectives on an Emerging Discipline,” Pearson, 1996.

[5] C. Hofmeister, P. Kruchten, R. Nord en H. Obbink, “Generalizing a Model of Software Architecture Design from Five Industrial Approaches,” *WICSA*, 2005.

[6] J. Tyree en A. Akerman, “Architecture decisions: Demystifying architecture,” *IEEE Software*, pp. 19-27, 22(2) 2005.

About CGI

Founded in 1976, CGI is among the largest IT and business consulting services firms in the world.

We are insights-driven and outcomes-based to help accelerate returns on your investments. Across 21 industry sectors in 400 locations worldwide, our 77,000 professionals provide comprehensive, scalable and sustainable IT and business consulting services that are informed globally and delivered locally.

cgi.com

CGI