**11. Enterprise IT Assessment for a Cloud Native Journey**

Shivakumar R Goniwada[**1**](https://learning.oreilly.com/library/view/cloud-native-architecture/9781484272268/html/511610_1_En_11_Chapter.xhtml#Aff2)

(1)

Bangalore, India

In the previous chapter, I explained how to modernize your legacy monolithic application to cloud native services by using decoupling, domain-driven design, and event storming.

Many enterprises fail or lose stream because they cannot demonstrate their business value to the customers and market. The modernization of a single system to cloud native technology does not show business value. So, you need to conduct an assessment that considers the following: application portfolio upgrade, technology modernization, optimization of the line of business, and the journey to the cloud to identify the maturity of the existing IT real estate.

An architecture assessment plays a vital role in both greenfield and brownfield application development, in the re-engineering of an existing application, and in the modernization and rationalization of portfolios.

Assessment is an activity to validate and review the existing IT real estate of an enterprise in terms of architecture, digital transformation, automation, software engineering, and cloud journey.

In this chapter, I will explain different ways to assess an enterprise by using various methods.

* What is an assessment?
* Different types of assessment
* Assessment strategic planning
* How to validate technical stability
* How to identify technology opportunities in an enterprise landscape
* Architecture assessment maturity model
* Digital transformation assessment maturity model
* Automation assessment maturity model
* Cloud transformation assessment maturity model

**Introduction**

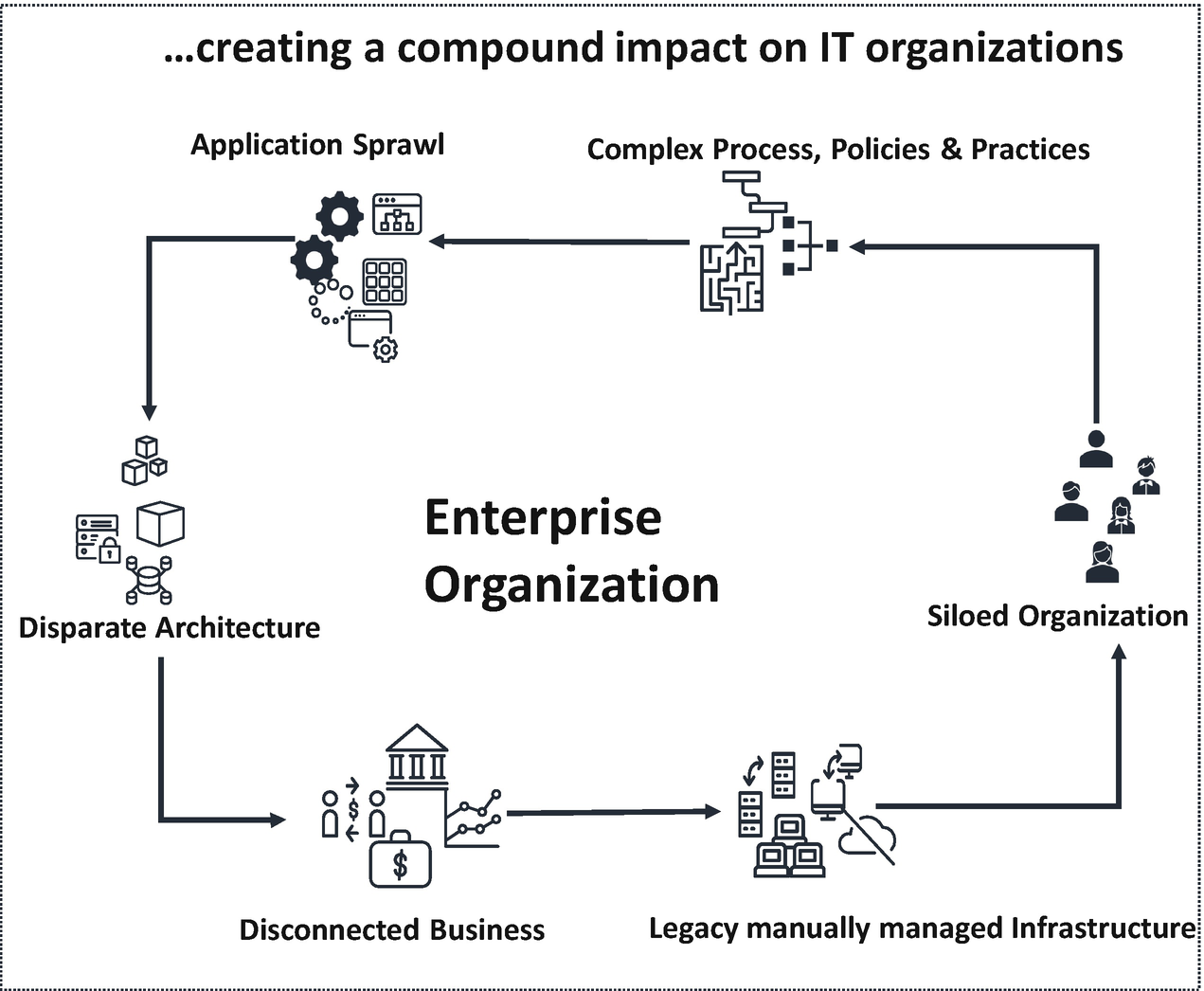
Digital-native companies across industries are fundamentally transforming every IT system to support business disruptions and embrace technological innovation. When confronted with large-scale digital disruption, enterprises must adapt quickly and lean into the cloud native by reimagining and elevating IT as a strategic core business function.

Large organizations are struggling to compete with new competitors and unicorns because of their own complex, tightly coupled IT real estate and aging technologies. Cloud native enterprises have invested heavily in cloud native technology platforms to create data-driven enterprises that provide business insight and that enhance the customer experience. Nimble new entrants like unicorns are leveraging new channels, scaling at an unprecedented pace, stealing market share, and rewriting the rules of the game by adopting a “culture of customer.”

Many enterprises across geographies with rigid legacy systems, siloed data, old economy workforce skills, and outdated operating models can’t compete with cloud native companies because emerging technologies scale quickly and can be delivered on demand by enabling machine learning and AI models. If you want to keep up with the digital disruptors, you must redefine IT by conducting as-is assessments. The assessment outcome provides a look at your landscape with recommendations. This helps you to strategize your transformation.

To reposition IT as a transformation engine, the leader of an enterprise should strive to change how IT engages with the business. For this journey, you need to know where your current problems are in the landscape and how to manage them.

Figure [11-1](https://learning.oreilly.com/library/view/cloud-native-architecture/9781484272268/html/511610_1_En_11_Chapter.xhtml#Fig1) illustrates why you need an assessment for your enterprise.



***Figure 11-1***

Disconnected enterprise organization

**Assessment**

An assessment provides a different way of conceiving, designing, and deploying technology across an enterprise. Assessment techniques help you to unlock unrealized streams of business value by optimizing your enterprise’s current IT real estate to innovate with a set of new technologies that are cloud native.

An assessment is a structured study on a well-defined set of IT capabilities or elements focusing on the technology and business aspects. It is guided by the agreed-upon objectives of the enterprise to analyze and document the as-is state and to make recommendations on the to-be state.

**What Is an Assessment Used For?**

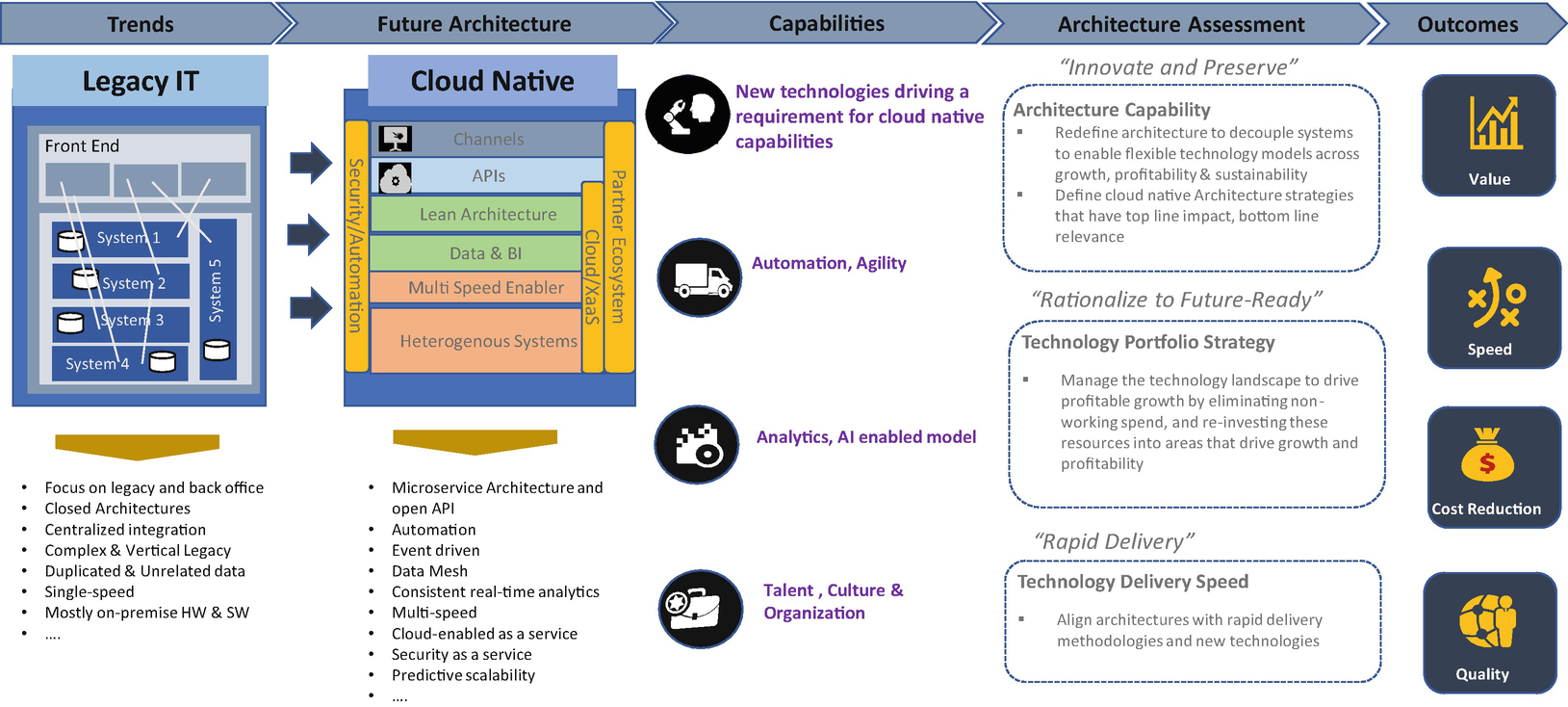
When you approach technology strategically, the core objective is to create business value for all aspects of the business, from the CxO to the back-office engineer and from software engineering to the customer experience. When you simplify your organization by using cloud native tools, you can better optimize your legacy estate.

There are multiple reasons to conduct an assessment.

* To prioritize resources to business objectives, continuously look for opportunities to create new business models, and transform experiences and industrialize operations quickly and efficiently with agility and automation.
* To build a future-ready IT foundation. The cloud native landscape is powered by the cloud, automation, and microservices to deliver customer value.
* To build more IT muscle for your business.
* To develop a new kind of system.
* To evaluate an architecture relative to best practices.
* To describe the structure and state of an architecture.
* To validate technical stability. For example, can the service scale instantly? Is the system able to self-heal?
* To identify technology opportunities, for example, cost reduction, agility, platform consolidation, portfolio rationalization, technology standardization, etc.
* To provide architecture for the due-diligence process, mergers and acquisitions, etc.
* To define the technology roadmap.
* To analyze the architecture’s dynamic behavior in response to external events.
* To provide details of value proposition, cost reduction, and decreased cycle time.

**Assessment Objectives**

The objective is to identify the widening gap between the business and IT capability through an architecture assessment. Another objective is to improve the customer experience, by making it faster and with higher quality, as illustrated in Figure [11-2](https://learning.oreilly.com/library/view/cloud-native-architecture/9781484272268/html/511610_1_En_11_Chapter.xhtml#Fig2).



***Figure 11-2***

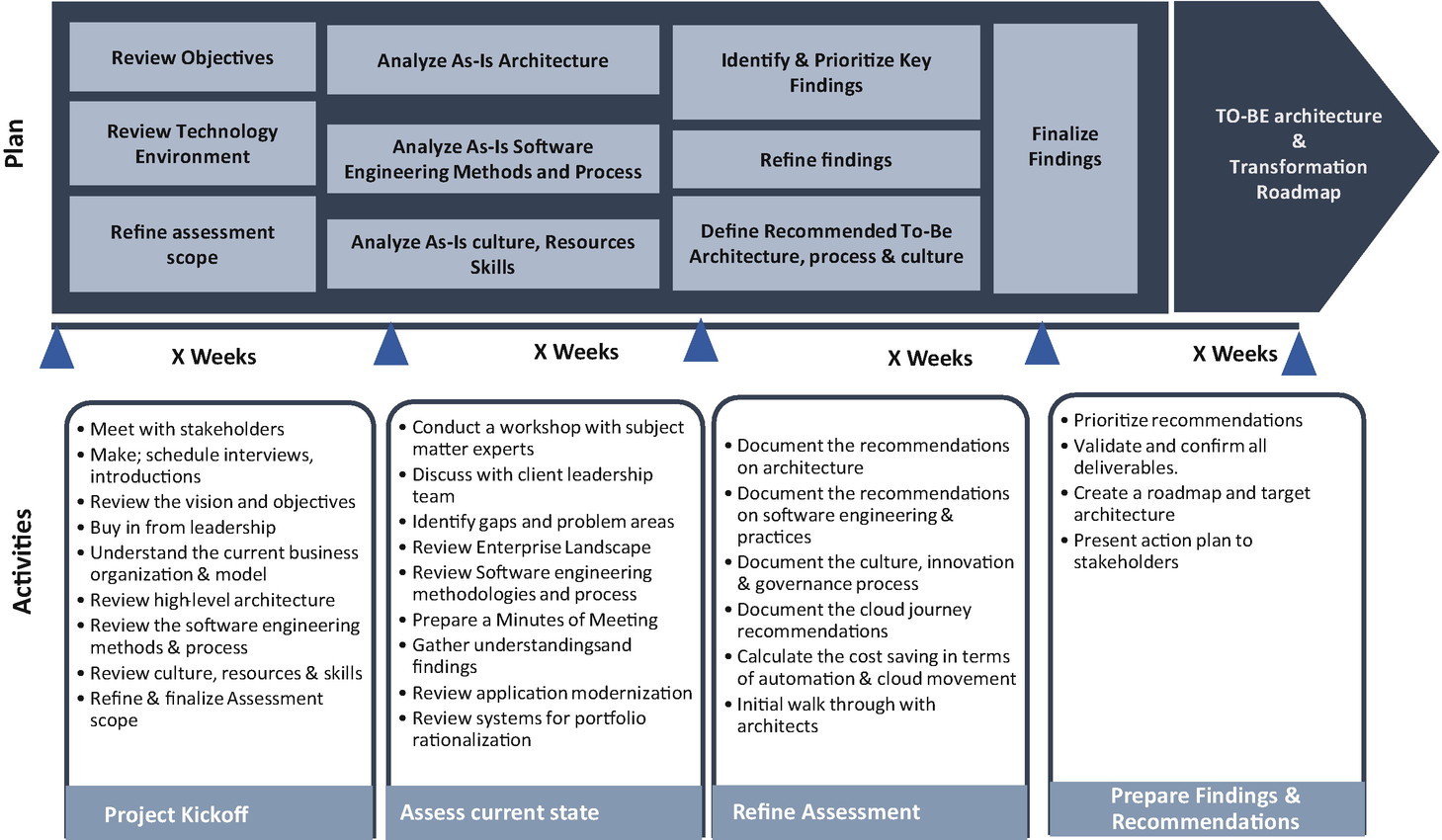
Assessment objectives

**Assessment Execution Approach and Key Activities**

What are the activities carried out during the assessment, and what are the deliverables of the assessment?

Based on the scope of assessment and type of assessment, you can decide how many weeks will be required to conduct an assessment and what key roles are required to conduct an assessment.

Figure [11-3](https://learning.oreilly.com/library/view/cloud-native-architecture/9781484272268/html/511610_1_En_11_Chapter.xhtml#Fig3) helps you to create a plan and deliverables for your client.



***Figure 11-3***

Assessment plan and activities

Along with the execution approach and activities, you need to have strong practitioners who have deep, relevant, and real-world experience. During the execution, you need to provide and run a delivery governance function to ensure the assessment is managed effectively and that the status and updates are effectively communicated throughout the delivery. During the entire assessment cycle, try to adopt critical governance communications by providing status reports to stakeholders.

You must adopt an agile approach for communicating findings and recommendations and follow these activities during the assessment:

* Hold an interim review /finding review session with the key identified stakeholders.
* Deliver formal outputs to the client team for review and distribution for early feedback.
* Create an execute summary pack for each workstream to give a high-level view of the approach, findings, and recommendations.

Along with this activity, you need to call out what you need from the client and who is required and how much time you require for workshops. The following sections cover different types of assessment.

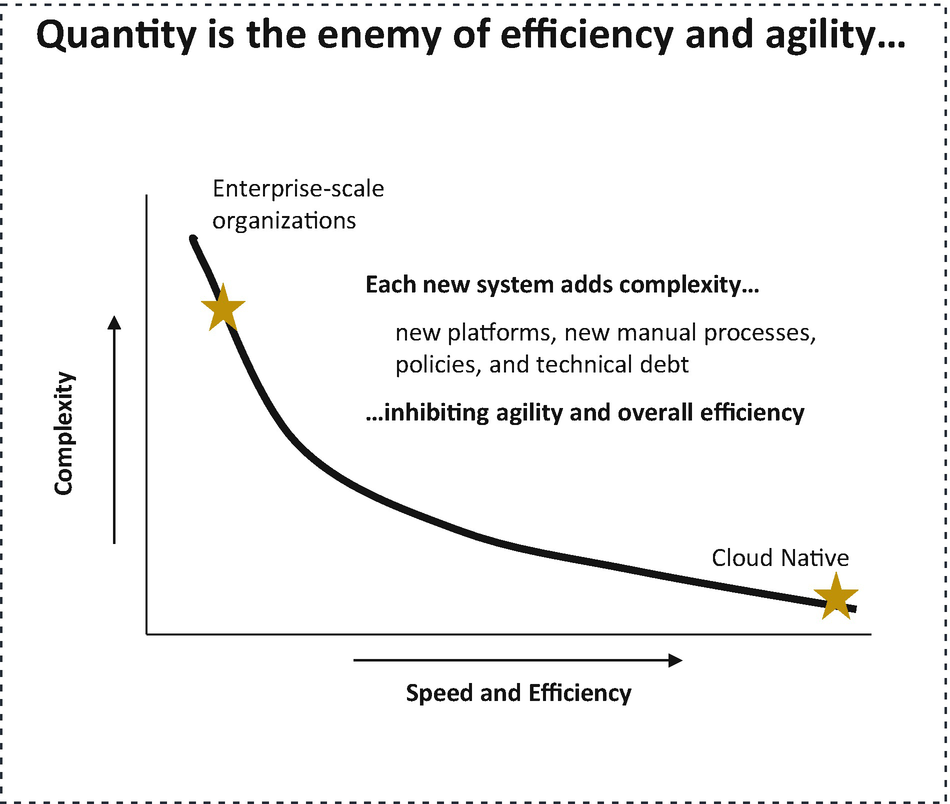
**Cloud Native Assessment**

The technologies and practices that have enabled success over the past decade are reaching the limits of their effectiveness. The next big practice is cloud native. To thrive, enterprises must design and execute cloud native technologies with unprecedented agility.

The cloud, microservices, containerization, eventing, serverless, and data meshes are all powered by cloud native. You can quickly identify promising new and emerging technologies that are relevant for your client landscape and scale them to deliver new capabilities, value, and business outcomes.

A cloud native assessment is more than assessing enterprises against new technologies; it helps your client to shift fundamentally from traditional technology strategy, development, and delivery models. This type of assessment uses modern architecture methods, tools, and techniques to drive a shift from project orientation to continuous development and the delivery of systems.

As shown in Figure [11-4](https://learning.oreilly.com/library/view/cloud-native-architecture/9781484272268/html/511610_1_En_11_Chapter.xhtml#Fig4), each system adds complexity to the cloud native journey. The y-axis represents speed and efficiency, and the x-axis represents complexity. The complexity of an organization increases while the speed and efficiency to cloud native reduces, and vice versa. Therefore, you need to choose the type and level of system for a cloud native journey.



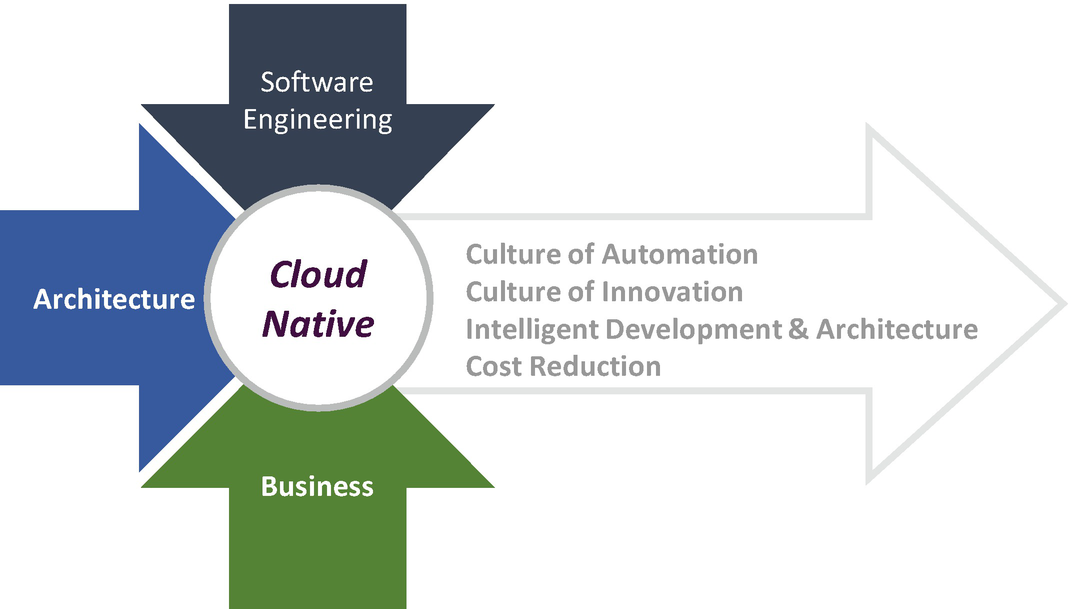
***Figure 11-4***

Cloud native assessment comparison

**When to Consider a Cloud Native Assessment**

As shown in Figure [11-5](https://learning.oreilly.com/library/view/cloud-native-architecture/9781484272268/html/511610_1_En_11_Chapter.xhtml#Fig5), in the cloud native assessment, you will assess the following:

* Methods and practices such as software engineering, automation, governance, deployment, etc.
* Technology and architecture such as cloud, microservices, event-driven, serverless, other cloud native elements
* Business alignment in terms of talent, innovation, culture



***Figure 11-5***

Cloud native assessment model

You can consider cloud native assessment for the following:

* *Responding to the strategic direction*: You see an immediate situation where your client is asked to improve time to market, reduce costs by optimizing various resources, and improve the quality by considering automation and a shift-left approach.
* *Enterprise team concerns*: Your client wants to change the way they are working and how they are working.
* *Enterprise estate*: An enterprise wants to move away from legacy monolithic applications, tight coupling, or poor delivery quality.
* *Manual delivery*: An enterprise wants to move away from old ways of working that result in manual builds, tests, and deployments that could be automated.

**Cloud Native Maturity Assessment Model**

A cloud native assessment is a model combined with quantitative analytics, gap assessments, and planning methodologies. It is the combination of existing practice area models, with a culture of automation and next-generation software engineering delivery concepts.

For each question in a cloud native maturity assessment, you must provide a rating based on the current practices and make recommendations based their maturity. The ratings are as follows:

* *Ad hoc*: With this rating, the process is not standardized, the architecture is not streamlined to cloud native, and agility is not followed. There is no vision available on cloud native adoption. This is like a one- or two-star rating.
* *Streamlined*: With this rating, the culture of automation exists but is not available across enterprises. Teams have started adopting cloud journeys but only in nonproduction environments, and few teams are following the agile approach with all 12 agile principles. This is like a three-star rating.
* *Optimized*: With this rating, a culture of automation exists but with only continuous integration, but there is no automation testing and no infrastructure automation. Few applications are deployed on the cloud, but they have started modernizing legacy monolithic applications to a cloud native architecture. This is like a four-star rating.
* *Matured*: With this rating, an agile pod approach is adopted across the organization, and most of the applications embrace cloud native architecture and collect real-time metrics. This is like a five-star rating.

You must prepare the questions for the cloud native assessment and evaluate each answer. Based on this evaluation, you will come to know where the current organization stands, and this will help you to make recommendations. One thing you need to remember, if the organization is ad hoc, don’t recommend the matured approach. None of the organizations can go from one to five stars quickly. They need time, and therefore make your recommendations wisely. Always provide multiple iterations before the organization reaches the maturity.

Table [11-1](https://learning.oreilly.com/library/view/cloud-native-architecture/9781484272268/html/511610_1_En_11_Chapter.xhtml#Tab1) shows some sample questions with data from a cloud native assessment.

**Note**

The score and comments are illustrative, and the data is from a global life science client. I conducted a cloud native maturity assessment for the entire enterprise. I interviewed all the relevant stakeholders including the leadership architecture team, project manager, product owner, development team, and infrastructure team. I captured these details during my workshop with the global life science client.

***Table 11-1***

Cloud Native Assessment Sample Questions

| Measure | Ad Hoc | Streamlined | Optimized | Matured | Current Score | Desired Score | Comments |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Cloud Native Architecture Roadmap** | There is no cloud native architecture roadmap. | There is a cloud native architecture roadmap that has some basic elements needed to build new capabilities. | A cloud native architecture roadmap is available and loosely followed. | A well-defined cloud native architecture roadmap is available, and teams across the organization have adopted it. | 2 | 3 | The organization has started a cloud native architecture journey and started creating a roadmap. |
| **Hypothesis-Driven Development (HDD)** | Teams uses observations and interviews to understand users but solely on quantitative studies. | Team follows HDD and uses both quantitative and qualitative analysis. | Team follows HDD and uses both quantitative and qualitative analysis. | Team follows HDD and regularly uses human-centered methods; involves users throughout the lifecycle of a project. | 1 | 2 | Some projects follow waterfall, and some follow the agile approach. |
| **Methods & Process** | Project teams do not follow any specific methods or process. | Team has followed some methods, but agile methods and process are not followed. | Most projects have consistent agile methods. | Most projects have consistent agile methods with @Scale IT. | 2 | 3 | Some projects follow agile method but do not have product owners or detailed backlogs available. |
| **Standards & Guidelines** | Team does not follow any standards or guidelines. | A few guidelines are available, and some teams follow the guidelines. | Follows the guidelines and also uses the tools to document the guidelines. | Follows the guidelines and has a process to share them across enterprise. | 2 | 3 | Team follows the standard guidelines and procedures, but they are not updating regularly, and the entire team does not have visibility. |
| **Design Standards** | No design standards exist. | Design standards have been created, but they are in static nature, and only a few in the project can access to them. | Design standards are available, and the project team is able to access them. | Design standards are available and are able to be accessed across team for reusability instead of reinventing the wheel. | 1 | 2 | No design standards are available. |
| **Culture of Automation** | Team does manual development and deployment. | Team does continuous integration, but testing and deployment are manual. | Team follows continuous integration and delivery and captures a few software engineering metrics. | Enterprise uses automation, and well-defined metrics are collected with infrastructure automation. | 1 | 3 | Team uses an ad hoc approach; there is no proper CI and CD pipeline. |
| **Environment Strategy** | There is no enterprise-wide environment provisioning; all are manual with require-based approach. | There is adoption of the cloud by a few teams in the organization but only for nonproduction environments. | There is adoption of the cloud across organizations for nonproduction environments, but few applications are running on the cloud. | There is a clear strategy available to adopt the cloud, and most of the applications are already in the cloud; some are on the cloud journey. | 1 | 3 | A few projects are adopting the cloud, but the majority of teams follow the on-premises approach. |
| **Modernization** | Organization has a lot of legacy applications and is running in on-premises virtual machines. | Organization started an application modernization journey by adopting cloud native but not for all the applications. | Organization has well-defined modernization roadmap and started adopting all cloud native elements. | Organization is already on the journey of modernization by adopting all the elements of cloud native. | 1 | 3 | Found huge technical debt, and 80% of systems in the enterprise are legacy monolithic systems. |

**Detailed Architecture Assessment**

The detailed architecture assessment model provides quantitative analytics, gap assessments of detailed existing architecture, and planning methodologies. It is an instrumental technique to validate technical stability and to identify technology opportunities. It is an essential part of technology due diligence and technology roadmap definition.

The cloud native architecture assessment will be used to assess the maturity of existing enterprises against cloud native elements such as microservices, APIs, agility, automation, event-driven, serverless, and cloud, and the detailed architecture assessment will be used to assess the stability of the existing systems in terms of application architecture, integration architecture, the behavior of an application for the “-ilities,” and more.

The outcome of this assessment helps you to decide whether the system is able to support modern-day business and can consider lift and shift to the cloud environment or whether the system can be modernized by using the domain-driven design technique.

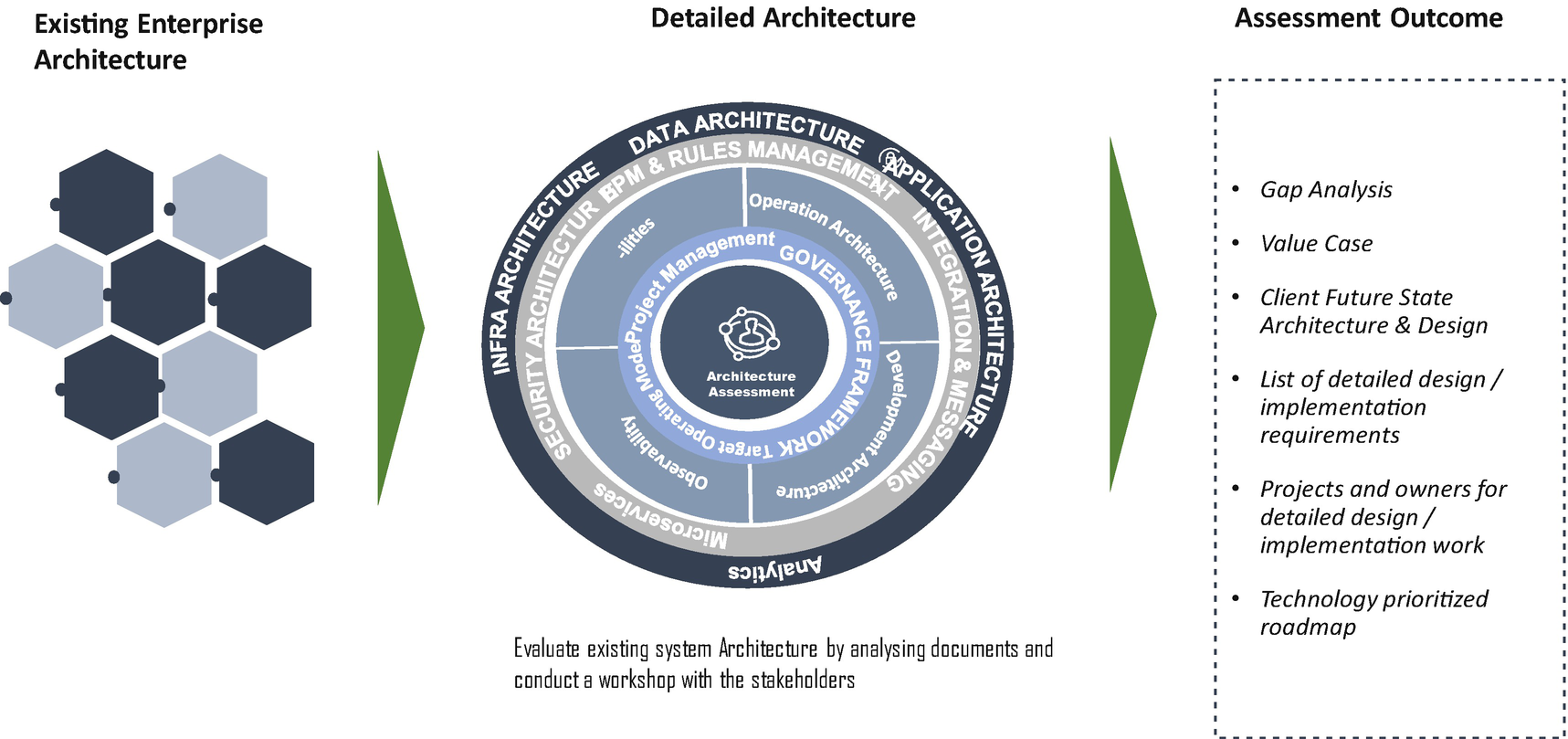
**Assessment Usage**

The main purpose of this assessment is to assess an existing behavior of a system with the capabilities of the application architecture, infrastructure architecture, development architecture, and “ilities” architecture. This assessment can be used to conduct a review at each system level or enterprise level across portfolios.

In this assessment, you need to prepare a questionnaire with questions that should be asked to drive the assessment such as interviews, workshops, and Q&A sessions with all the relevant stakeholders.

**Architecture Assessment Model**

Figure [11-6](https://learning.oreilly.com/library/view/cloud-native-architecture/9781484272268/html/511610_1_En_11_Chapter.xhtml#Fig6) shows the assessment model. The following assessment model capture the current results and analyze the document findings and provide recommendations .



***Figure 11-6***

Assessment model

In the planning phase of the assessment, you need to identify the scope of an assessment, for example, if its scope is for only one system or one portfolio/department or the entire enterprise. Identifying the scope drives the accuracy, complexity, and costs of an assessment. In the planning phase, you also need to define a clear objective, investigation area, and capabilities. Once the scope is identified, you need to define a plan for how many resources are required, what skills are required, and how much time it will take.

In the workshop phase of the assessment, you need to work with the client team to identify subject matter experts (SMEs), and the time required from them, and make sure the client stakeholders are committed to this. One thing you need to remember is that you need seriousness and active participation from client SMEs and other stakeholders for a successful assessment. Based on their availability, schedule a workshop with the relevant stakeholders. Note: invite only the relevant SMEs. It is better to share the list of assessment questionnaires with the SMEs in advance so they can prepare before the workshop.

In the capture content phase of the assessment, you collect existing documents and review each of them. You document every discussion of the functional capabilities of a system, the technology capabilities of a system, the architecture styles used, the pain points, the software engineering capabilities of a system, the behavior of a system, etc. You need to conduct a thorough investigation by going through the questions of each capability and document answers and references by using the issues, risks, opportunities, and strengths (IROS) model and prioritize and group them based on the capabilities.

In the recommendation phase of the assessment, evaluate the findings based on priority and business importance, and produce an assessment report. The assessment report must contain only the official recommendations of findings. Usually, the final report contains an assessment scope overview, scope element description, assessment findings, prioritization of each finding, roadmap, reference implementation, best practices, and conclusion.

**Assessment Questions Template**

You must prepare the questions on a detailed architecture assessment for each capability and rate each answer. The rating can be from 1 to 5 based on current industry and technology trends, and you can document the client feedback on each question.

For example, let’s consider the rules management question, “Are you using rules management, and if yes, how are the rules managed?” If the client feedback is yes, then you need to probe further. What kind of rules? Are they using only action rules or decision rules? Are the rules are externalized? Are they using any ML-based model for rules? If they answer yes for all, then your rating will be 5. If they are using rules management and externalizing the rules, then your rating will be 4. If they are managing rules externally in the configuration, fine; then it should be 3. If they are hard-coding the rules in source code, then it is 1 or 2.

Similarly, you need to probe the client by asking more questions until the end.

You can find a few sample questions for the detailed assessment in Table [11-2](https://learning.oreilly.com/library/view/cloud-native-architecture/9781484272268/html/511610_1_En_11_Chapter.xhtml#Tab2) and prepare similar ones for a full-fledged assessment.

**Note**

The score and comments are illustrative, and the data is from a global retail client. I conducted a detailed architecture assessment for a portfolio. I interviewed all the relevant stakeholders including the architecture team, project manager, product owner, development team, and infrastructure team. I captured these details during my workshop with the global retail client.

***Table 11-2***

Detailed Architecture Assessment Sample Questions

| Capability | Question | Client Feedback | Current State (Rating) | Target State (Rating) | Best Practices | Comments |
| --- | --- | --- | --- | --- | --- | --- |
| **UI Architecture** | What are the UI architecture components, and how do they communicate to the back-end system? | Some projects are designed with responsiveness, but the majority of applications are in old legacy technologies. | 1 | 3 | No best practices are followed. | Only a few projects use the latest technologies with responsive design, and the majority of the applications are client-server architecture. |
| **UI Architecture** | What technologies are involved in both web and mobile applications? | Web applications are using legacy technologies, and a few systems use responsive design. | 2 | 3 | Best practices are available, but teams are not following them properly. | There is no roadmap to adopt a responsive design. |
| **Rules Management** | Are you using rules management? If yes, how are rules managed? | Rules are part of systems. | 1 | 3 | No best practices are available. | Rules are not defined properly, and rules are not externalized; a few systems require frequent changes to update rules. |
| **BPM** | Are you using a business process in your application? If yes, is this automated? | Few systems use the BPM tools. | 1 | 3 | Following a few best practices. | Systems are legacy, and there is no enterprise-wide approach. |
| **Integration** | What kind of integration are you using? Are you using APIs or event-based integration? | Few systems are APIs, and some systems use an MQ-based approach. | 1 | 3 | API standards are not available. | The APIs are available without any standards, and teams are using MQ for point-to-point. |
| **Data Integration** | How is data integrated across enterprises? Are you using any ETL or CDC approach? | All integrations are batch. | 1 | 3 | ETL standards are followed. | Teams want to move into real-time streaming from batches, but a roadmap is not available. |
| **Data Architecture** | How is the data stored in your system? How are transactions managed? | All systems are using monolithic database. | 1 | 2 | Data standards are available. | Teams are using a single monolithic RDBMS. |
| **Software Engineering** | What methodology should you follow for development of a system? Are you using any DevOps pipeline? | Yes, few projects are using CI but no CD. | 2 | 3 | No best practices are available. | Teams are following only CI, but testing and deployment are manual; client wants to automate end-to-end lifecycle. |
| **Software Engineering** | How can you conduct testing? Are you using any automation frameworks or doing it manually? | Executing manual testing. | 2 | 3 | Test cases are available; using manual test data. | Teams are conducting manual testing for every release and need to automate CD as part of delivery. |
| **Software Engineering** | How are the issues and risks managed? | Uses Microsoft Excel. | 1 | 3 | Follows organization best practices. | Uses Excel but wants to automate risks and issues in AML tools. |
| **Operation Architecture** | How are you managing applications? Are you using bots to improve the operation capability? | All operations are manual. | 1 | 2 | Operation procedures are followed. | First wants to streamline operations and later wants to use AIOPs. |
| **Operation Architecture** | Are you using any monitoring tools? If yes, what kind of tools? What are the metrics you collect? | Some systems are using monitoring. | 1 | 3 | Ad hoc; no standard available. | Few systems use the monitoring tools and capture the metrics, traces, etc., but the majority does not follow them. |
| **-ilities** | How is application and perimeter security is managed? | Security is very well established. | 2 | 3 | Yes, they have followed good security practices. | Follows security but wants to improve with the latest tools and configuration. |
| **-ilities** | Are there any pain points in the system behavior at runtime? | Yes, company is facing lot of issues in production environment. | 1 | 3 | No best practices are followed. | The client wants to automate the system bottlenecks and self-healing mechanism. |
| **-ilities** | How are you managing peak load? Is the system highly available? if yes, how many nines? | Majority of the systems are using on-premises. | 1 | 3 | No best practices are followed. | The majority of systems are legacy and deployed on-prem; wants to start cloud journey. |
| **-ilities** | Tell us about application performance. Are you facing any performance glitches? | Some systems behave very well, and some are unable to scale. | 1 | 2 | No best practices for performance tuning. | Wants an automated performance management. |
| **Infrastructure Architecture** | How is your application deployed? Is it deployed in VMs or containers? | 90% of applications are VMs. | 1 | 2 |  | Few systems are deployed in a container; wants to improve container adoption percentage. |
| **Application Architecture** | Tell us about the application architecture of your system. What are the technologies used? | Legacy systems; some are very old. | 1 | 3 | No best practices are available. | Old legacy monolith applications, and some systems developed ages ago; wants to embrace cloud native. |
| **Project Management** | How is your application delivered? What kind of metrics do you collect? How do you report to leadership? | Some projects follow agile and waterfall and capture delivery metrics. | 2 | 3 | Delivery best practices are available; some are ad hoc. | Team wants to embrace agile and pod culture. |

**Automation Maturity Assessment**

Automation is an essential part of every organization for a cloud native journey. Based on recent Gartner research, 39 percent of an organization wants to improve automation strategy and innovation, 23 percent wants to develop a stronger talent model, 23 percent feels their team lacks an understanding of automation trends, and 11 percent feels resistant to change.

There is some kind of automation that exists in almost all enterprises around the globe; However, these enterprises do not know whether they are following the right automation approach, what the gaps are compared to industry standards, etc. This maturity assessment helps you to find the client’s concerns and the level of maturity of automation.

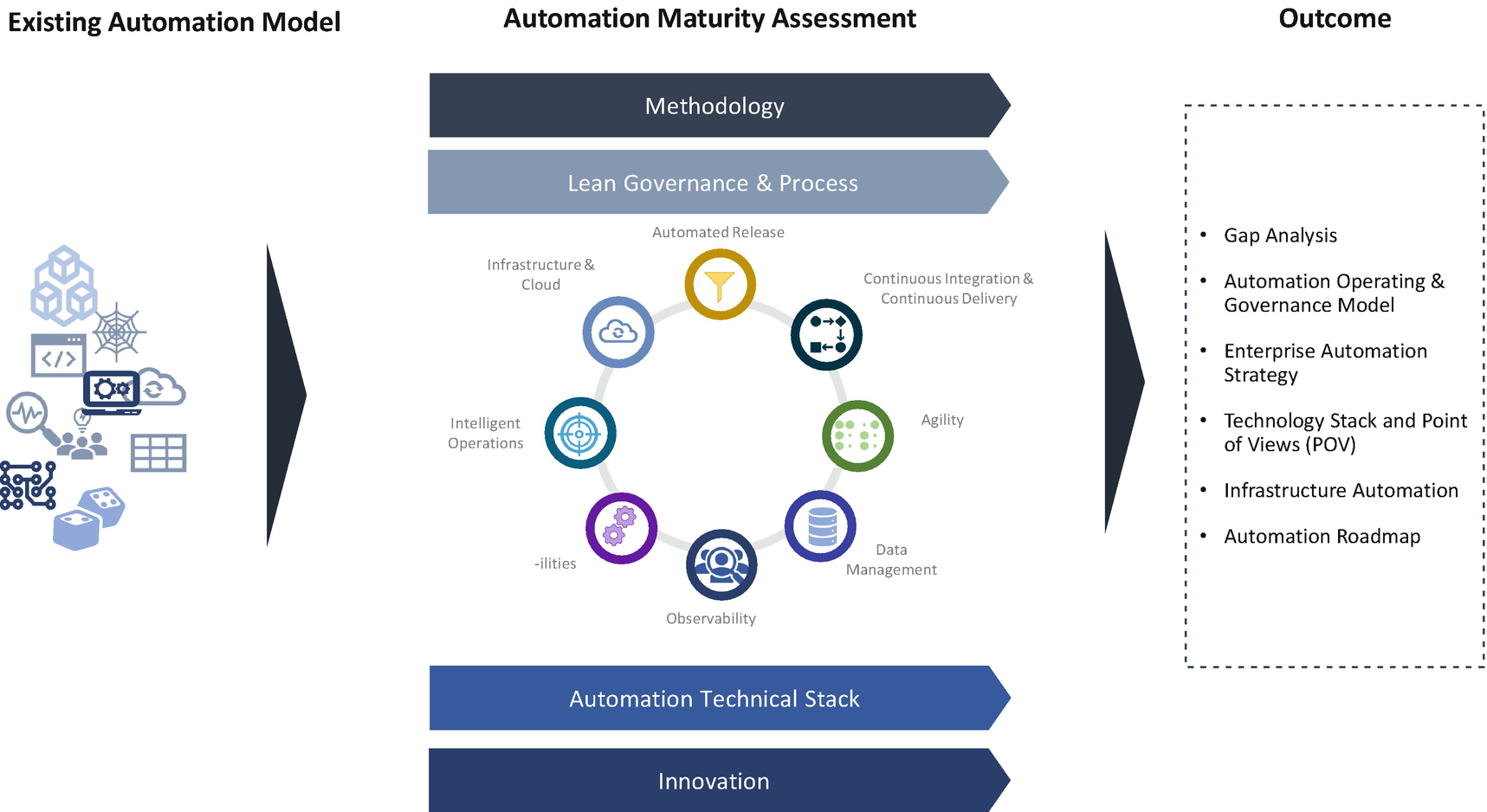
The automation maturity assessment framework is to assess the automation maturity of the organization to align with the enterprise’s vision and industry standards. It will help you to understand the current gaps and identify a case for change.

To achieve the true potential of automation, you need to look for opportunities across the enterprise and cover every project of an organization. Automation can achieve the following:

* *Enhance user experience*: Customer satisfaction with the timely availability of data.
* *Optimize process*: Identify relevance in the current market and quick responsiveness, and optimize process components with the right process elements.
* *Drive cloud native journey*: Identify applications to connect better with the customer, and automate physical and virtual environments to support cloud native and shift left on security vulnerability.
* *Focus on generating revenue*: Reduce repetitive, standard work and enhance cost, efficiency, and reliability.

**Automation Maturity Assessment Model**

The maturity assessment model is a combination of capabilities, process, governance, technical stack, and performance management that enables the enterprise to deliver services to its customers. In the automation model shown in Figure [11-7](https://learning.oreilly.com/library/view/cloud-native-architecture/9781484272268/html/511610_1_En_11_Chapter.xhtml#Fig7), evaluate the current environment using the diagnostic and maturity model and create value cases for the automation journey. This model helps you to identify the impact of change assessment and the impact on people, process, skills, organization, and culture.



***Figure 11-7***

Automation maturity assessment model

To conduct a maturity assessment, you follow a similar approach as illustrated in the previous assessment types, like plan for workshops, capture the current state, analyze and document findings, and provide recommendations.

During the engagement, review the automation operating model’s current state for the key technology elements, adoption of technologies, skills and talent availability, and change management model.

**Automation Maturity Assessment Questionnaire Template**

You must prepare the questions on automation maturity assessment for each capability and evaluate the answers. The rating can be ad hoc, streamlined, optimized, and matured (refer to the earlier cloud native assessment for details). Then document the client feedback on each question.

You need to consider the following areas for a maturity assessment:

* *Delivery organization and methodology*: In this area, conduct an assessment of release frequency to production and nonproduction environments, delivery organization structure, delivery and process, and metrics collection.
* *Delivery governance and process*: In this area, conduct an assessment on governance approach and product management.
* *Automated release software*: In this area, conduct an assessment on source control and binaries management, deployment process, and change request management.
* *Continuous integration and delivery*: In this area, conduct an assessment on pipeline, testing, and reviews.
* *Infrastructure automation*: In this area, conduct an assessment on infrastructure as a code, containerization, and security configuration.
* *Intelligent operation*: In this area, conduct an assessment on observability, knowledge management, and ticket management.
* *Innovation*: In this area, conduct an assessment, new process adoption, culture of automation, and culture of innovation.

You can find a few sample questions for an automation maturity assessment in Table [11-3](https://learning.oreilly.com/library/view/cloud-native-architecture/9781484272268/html/511610_1_En_11_Chapter.xhtml#Tab3); use them as a jumping-off point to prepare similar ones for a full-fledged assessment.

**Note**

The score and comments are illustrative, and the data is from a global retail client along with cloud native journey. I conducted a detailed architecture assessment for a portfolio. I interviewed all the relevant stakeholders including the architecture team, scrum manager, product owner, development team, and infrastructure team. I captured these details during my workshop with the global retail client.

***Table 11-3***

Automation Maturity Assessment Sample Questions

| Area | Ad-hoc | Streamlined | Optimized | Matured | Current Score | Desired Score | Comments |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Delivery Organization and Methodology: Release Frequency** | Release process is poorly defined, with ad hoc change requests and long gaps between releases. | Release process is managed and has had multiple postponements of releases. | Release process is well-defined, and product backlog is stable. | Releases on demand; have a very good process; even small changes can be pushed to environments. | 1 | 3 | A few projects follow waterfall, and a few projects follow agile, but there is no backlog available; wants an integrated backlog. |
| **Delivery Organization and Methodology: Process** | Process and methodology are not defined. | Follows waterfall methodology. | Follows a mix of waterfall and iterative approach. | Follows agile with pod culture. | 1 | 3 | Wants to embrace both waterfall and agile with a streamlined approach. |
| **Delivery Governance and Process: Governance** | Governance process does not exist. | Delivery governance process is based on a traditional approach like waterfall etc. | Centralized governance approach. | Centralized with decentralized approach like federated approach. | 1 | 3 | Ad hoc governance and decisions are based on resources, not based on roles. |
| **Delivery Governance and Process: Product Management** | No product road map exists; follows ad hoc management. | Product management exists but limited to certain extents like defects, etc. | Product owner is empowered to make decisions. | Long-term roadmap available and fully controlled by product owner. | 1 | 3 | There is no roadmap for a project. |
| **Automated Release Software: Source Management** | Code merge is manual, and no proper management exists. | Code and binaries must be version controlled, and merge is semi-automatic. | All code and binaries are versioned by using tools, and auto merge exists. | Follows well-defined branching strategy; traces user stories and able to track. | 1 | 3 | Code merges are manual, and some systems and managing branches properly want to embrace feature branching. |
| **CI/CD: CI** | CI pipeline exists, but there are no proper jobs. | CI pipeline exists; has proper jobs but no automated review mechanism. | CI pipeline exists with all required jobs with review mechanism and quality gate. | CI pipeline exists with all jobs and follows the shift-left approach. | 1 | 3 | CI exists, but for few projects; wants to adopt cloud native best practices. |
| **CI/CD: CD** | Ad hoc manual testing. | Test scripts are defined and executed. | Automated test script available and executed as part of pipeline. | All tests are automated and run for all environments and run in live estate. | 1 | 3 | All testing is manual and wants automated approach. |
| **Infrastructure Automation: Automation** | Deployment of artifacts are done manually. | Artifacts deployment by using auto scripts. | Infrastructure configuration are done automatically by using infrastructure as code. | Configured with infrastructure as code across multicloud vendors. | 1 | 3 | All infrastructure is manual; wants to do MVP for few projects before embracing across organization. |
| **Intelligent Operation: Observability** | No monitoring tools available. | Tools in place, but not all application are configured. | Well-defined monitoring tools are available and configured properly. | Well-defined integrated monitoring; follows the observability as a service. | 2 | 3 | Tools are available but not captured with metrics, etc. |
| **Innovation** | Organization does not have roadmap on automation. | Few projects in an organization use automation; skills exist. | Well-defined roadmap available, and team follows the culture of automation. | Whole enterprise adopted the culture of automation and culture of innovation. | 1 | 3 | There is no enterprise-wide automation available and exists for only a few projects. |

**Summary**

In this chapter, I explained different kinds of assessment and evaluation models and showed examples with illustrative data. These questions are sample ones that you need to add depending on the type of client and nature of assessment. By using these frameworks, you can evaluate the IT real estate maturity and provide recommendations with a clear roadmap for both tactical fixes and the strategic journey.

The following are the best practices when conducting an assessment:

* Find out the client’s and stakeholder’s commitment.
* Choose the right SME with deep skills.
* Interview the right stakeholders and pose the right questions.
* Capture all the details and take minutes at the meeting.
* Evaluate each question and rate them, and then prepare observations and recommendations.
* For each recommendation, provide industry best practices and references.
* Prepare a roadmap for separating tactical and strategic goals.
* Review your recommendations internally.
* Arrange regular meetings with a small group of stakeholders to discuss your observations and recommendations.
* Finally, read the final report to a larger audience.

Similarly, you can assess the client API maturity, software engineering approach, etc.