



EBOOK

The Impact of Continuous Testing

How Organizations Transform Their Testing from Reactive to Innovative

Continuous testing is the preferred model for teams looking to create, test, and release high-quality applications today. If you are involved in the application development landscape, you know how important continuous testing is to ensure your applications not only meet expectations but exceed them.

Yet not all continuous testing strategies are created equal. Depending on where your team is in their level of continuous testing implementation—and depending on where your application is in the development lifecycle—your approach to continuous testing may vary.

This is where continuous testing maturity models come in.

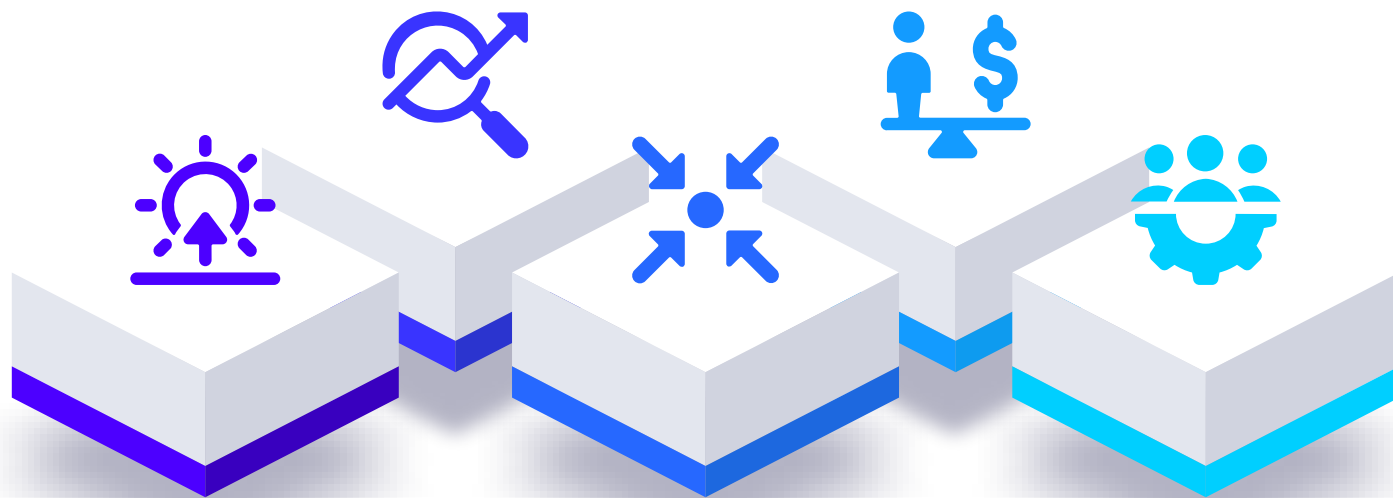
This guide will delve into continuous testing maturity models and their impact on different areas of business. It will also illustrate how adopting these continuous testing maturity models can help take your organization’s strategy from reactive to innovative, paving the way for more successful applications that don’t just meet expectations but exceed them.

What is Testing Maturity?

Testing maturity identifies where in the testing process steps and methods can be improved upon. It is an indicator of how the testing is designed, measured, managed, monitored, and how the results are yielded.

The benefit of using a testing maturity model is to continuously refine and improve upon your team’s testing strategy to get the best application you possibly can. Without self-evaluation, you run the risk of your testing methods becoming stagnant and archaic, which will negatively impact the quality of your app.

Just as the integrity of your testing is important for the quality of your app, so too is the structure of your testing maturity model important to the quality of your testing. Which brings us to setting up and establishing your model based on a few key pillars.



The 5 Pillars of a Continuous Testing Maturity Model

Think of your testing maturity model as a two-phased approach — the Capability phase and the Assessment phase. We will discuss the Assessment phase later, so for now let us focus and break down the five key components also known as pillars of the Capability phase. As the name suggests, the goals will vary depending on which level of capability your testing is at.



Pillar 1: Initial

This entry level pillar is to make sure the software runs successfully. This is the bare bones testing strategy, which lacks experienced testers, testing tools, or necessary resources. There are no testing landmarks or processes established. Does the software work? Yes? Ship it — no QA checks are done prior.



Pillar 2: Defined

The next step to take towards achieving testing maturity is to establish debugging goals and policies. This means separating and delineating testing and debugging to be conducted with intention. Testing should be done after coding to determine if the software is meeting specifications at the project's outset. This phase is also known as the Basic phase.



Pillar 3: Integrated

This pillar will see testing integrated into the software development lifecycle (SDLC). Tests become more organized, and objectives are determined based on priority. Testing is now considered an essential aspect of software development. This phase is also known as the Automated phase.



Pillar 4: Management & Measurement

How do you know your tests are effective and doing what they are supposed to? Implementing key measurements and quantifications of what a successful test looks like is the next step in your testing maturity model. Reviews at each stage of the development process are treated as tests. Regression testing becomes incorporated to collect and record in a database for reference. Bugs are identified, logged, and given security levels based on severity. This phase is also known as the Orchestrated phase.



Pillar 5: Optimization

Now is the time to determine how well your testing process is managed and defined. You should be tracking not only test effectiveness but also how it impacts your budget. QA and deflection prevention (as opposed to detection) are now used. The team now has tools to support testing metrics, as well as test design and defect gathering. This phase is also known as the Intelligent phase.

Continuous Testing Maturity Models

Now that we have covered the 5 pillars of a continuous testing maturity model, let us take a look at some examples of maturity models in action.

Functional Testing

Functional testing is the process of validating the functionality of a software application. Functional testing assesses whether the primary functions of a software application are working and that they align with the overarching goals for the app. Functional tests result in 'pass' or 'fail' scenarios because the features either work as designed or they do not.

Some examples of functional testing types include Unit Testing, Component Testing, Integration Testing, End-to-End Testing, Sanity Testing, and Smoke Testing.

Functional Testing Maturity Model	PHASE 1 Initial	PHASE 2 Reactive	PHASE 3 Proactive	PHASE 4 Optimizing	PHASE 5 Innovating
KPI: Platform Coverage	1 of each platform	15-29%	30-49%	50-79%	80-100%
KPI: Automation %	~10%	~33%	~66%	~75%	80%+
KPI: Automation Stability	0-29%	30-69%	70-95%	95%+	99%+
Quality Gates	Test new features only End of sprint only	Add in-sprint testing	Add daily smoke Add nightly regression	Add scheduled sanity Add smoke on every PR	Add full regression on every PR
Business Value	Biannual / quarterly release	Release every 2 months	Monthly release	Biweekly release	Daily
How Perforce Can Help	Professional Services Establish basic lab	Professional Services Increase lab size	Professional Services - Increase lab to meet desired feedback window Scriptless Mobile	Professional Services Introduce virtual lab to speed up feedback	Professional Services SV and TD - required to shift left and perform component level testing advanced testing introduced

The chart depicts the different phases (or pillars) of a functional testing maturity model.

Take automation, for instance. As you can see, in Phase 1 (or the Initial phase), only around 10 percent of the example is automated, while in Phase 5 (or the Innovating/Optimization phase), at least 80 percent is automated.

In the Initial phase, teams are only testing new features and end of sprint only, while they can test a full regression on every pull request by Phase 5.

Load Testing

Load testing measures how well an application performs under real-world conditions, such as peak traffic events like Black Friday for retail apps. Load testing ensures that your application can handle greater levels of usage and helps identify areas of potential weakness in advance.

The chart on the next page depicts the different phases (or pillars) of a load testing maturity model.

You will see that in Phase 1 (or the Initial phase) of the Focus of Testing, load testing is done in an ad-hoc manner, with production defects often getting reproduced. By Phase 5 (or the Innovating/Optimization phase), load testing has become fully optimized to the point where the team is pushing testing boundaries, performing chaos testing and UX testing to ensure the best possible application quality.

When measuring the business value in this load testing maturity model, in the Initial phase end users are already impacted and the focus is merely on damage control. By the Optimization phase, the ROI is achieving best-in-class applications.

Load Testing Maturity Model	PHASE 1 Initial	PHASE 2 Reactive	PHASE 3 Proactive	PHASE 4 Optimizing	PHASE 5 Innovating
KPI: Focus of Testing	Ad-hoc Performance Testing, Reproduce production defects	Test critical path before major events and Performance Testing late in SDLC	Establish test suite and integrate to CI/CD and isolation/ component level testing as part of process	Optimize test suite, leveraging feedback from analytics	Pushing the boundary of testing
KPI: Performance Triage Response	Multiple sprints and hot fixes	Defects identified end of cycle, mostly backlogged and work on prioritized items only	Leverage SLA and benchmarks to identify performance regression during QA/ UAT builds	Leverage SLA and benchmarks to identify performance regression within development cycle	Developers have the ability to triage before PR
Quality Gates	Developer station only	Add test on basic QA environment; Establish early performance baseline	Add test on production like QA environment; Add component level test; Add Integrated into CI/CD pipelines	Add Production testing	Add component level testing as part of PR process
Business Value	End users already impacted. ROI - Focus on damage control	ROI - Strategy is focused on risk mitigation, often leading up to critical events such as Black Friday sales	ROI - Focus on reducing responsiveness to performance incidents and increasing developer productivity	ROI - Focus on increasing developer productivity and driving efficiency into the process	ROI - Weaponizing development as a differentiator, achieving best in class
How Perforce Can Help	Load Testing Professional Services	Load Testing Professional Services	Load Testing Service Virtualization Test Data Professional Services	Load Testing Service Virtualization Test Data Professional Services	Enterprise License Agreement

The Impact of Continuous Testing Maturity Models

Now that we have delved into the details of functional and load testing maturity models, we will take a closer look at the impact continuous testing maturity models can have on different areas of business, starting with the business impact itself.

Business Impact and Metrics Summary



Business Impact – Initial Phase

In terms of business impact, the Initial phase of a continuous testing maturity model involves unpredictable quality and high costs. Manual testing is the norm at this stage, with minimal automation. Rather than operating under the continuous testing model that gives every team member a role in ensuring quality throughout the application development lifecycle, quality assurance is placed on individuals or specific SMEs—a much less efficient and thorough approach that leads to poor and unpredictable quality. The measures of test and data coverage during the initial phase are often un-scientific. These inefficient processes and lack of continuous testing lead to greater waste and higher costs.

Business Impact – Basic Phase

The Basic phase can be defined by reduced costs and improved code quality when compared to the Initial phase. It is also marked by an increase in test coverage, improved test environment setup time, and an increased effort in test design and test maintenance efforts. Yet, there are often defects that still leak through to production (though they have decreased) as well as an ambiguity in some testing requirements.

Business Impact – Automated Phase

In the Automated phase, there is an improvement in development and test quality as well as an improvement in overall app quality. Shift left testing activities have been embraced, test automation has significantly increased, and test execution time has gone down.

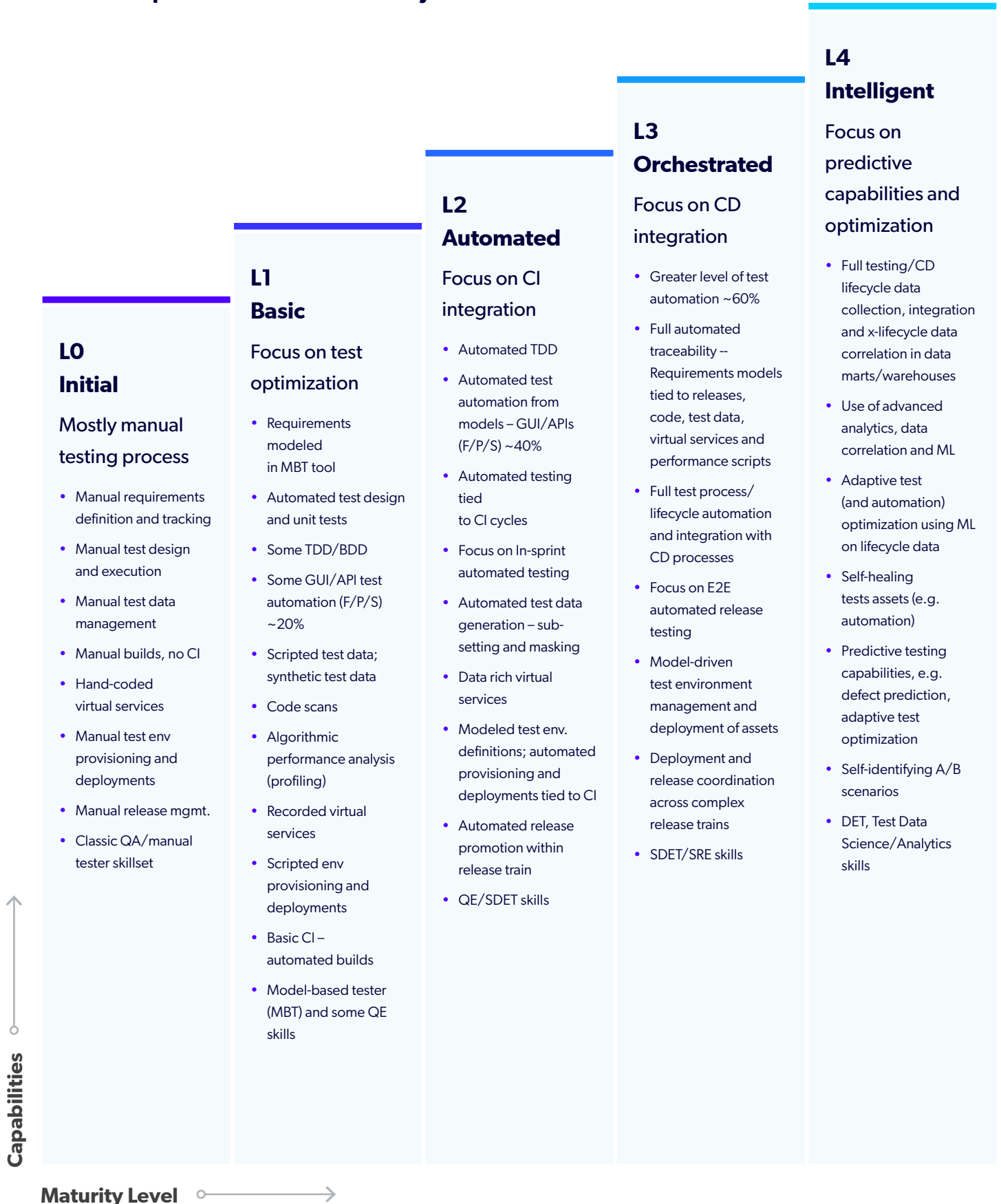
Business Impact – Orchestrated Phase

This phase is marked by an overall improvement in application release agility. Nearly everything is now automated, test execution time has significantly improved, and continuous testing has become a key part of the DevOps pipeline. In addition, leaked defects are very rare during the Orchestrated phase.

Business Impact – Intelligent Phase

The final phase is marked by optimization and predictability. Defects are prevented, there is a robust test scoping and maintenance effort, and tests are properly prioritized. At this final stage of the continuous testing maturity model, businesses can trust in their process without worrying about quality assurance, tests are fully automated, and resources are allocated properly — ensuring the best quality applications and therefore the best possible business outcomes.

Process Impact and Metrics Summary



Process Impact – Initial Phase

In the Initial phase of process maturity, the testing process is mostly manual—from manual requirements and tracking to manual test design and execution and manual test data management.

Virtual services are hand-coded and test environments are provided and deployed manually. In addition, this phase includes manual release management utilizing a classic QA/manual tester skillset.

Process Impact – Basic Phase

In the next phase, there is a greater focus on test optimization. Test design and unit tests are generally automated, there is around 20 percent API test automation, as well as some use of scripted and synthetic test data. During this phase there will likely be a model-based tester (MBT) and some team members with some QE skills.

Process Impact – Automated Phase

The Automated phase of process maturity is marked by a focus on CI integration. At this stage, API test automation is around 40 percent, automation testing is tied to CI cycles, there is a focus on in-sprint automated testing, and data rich virtual services are utilized. There will be QE/SDET skills at this phase.

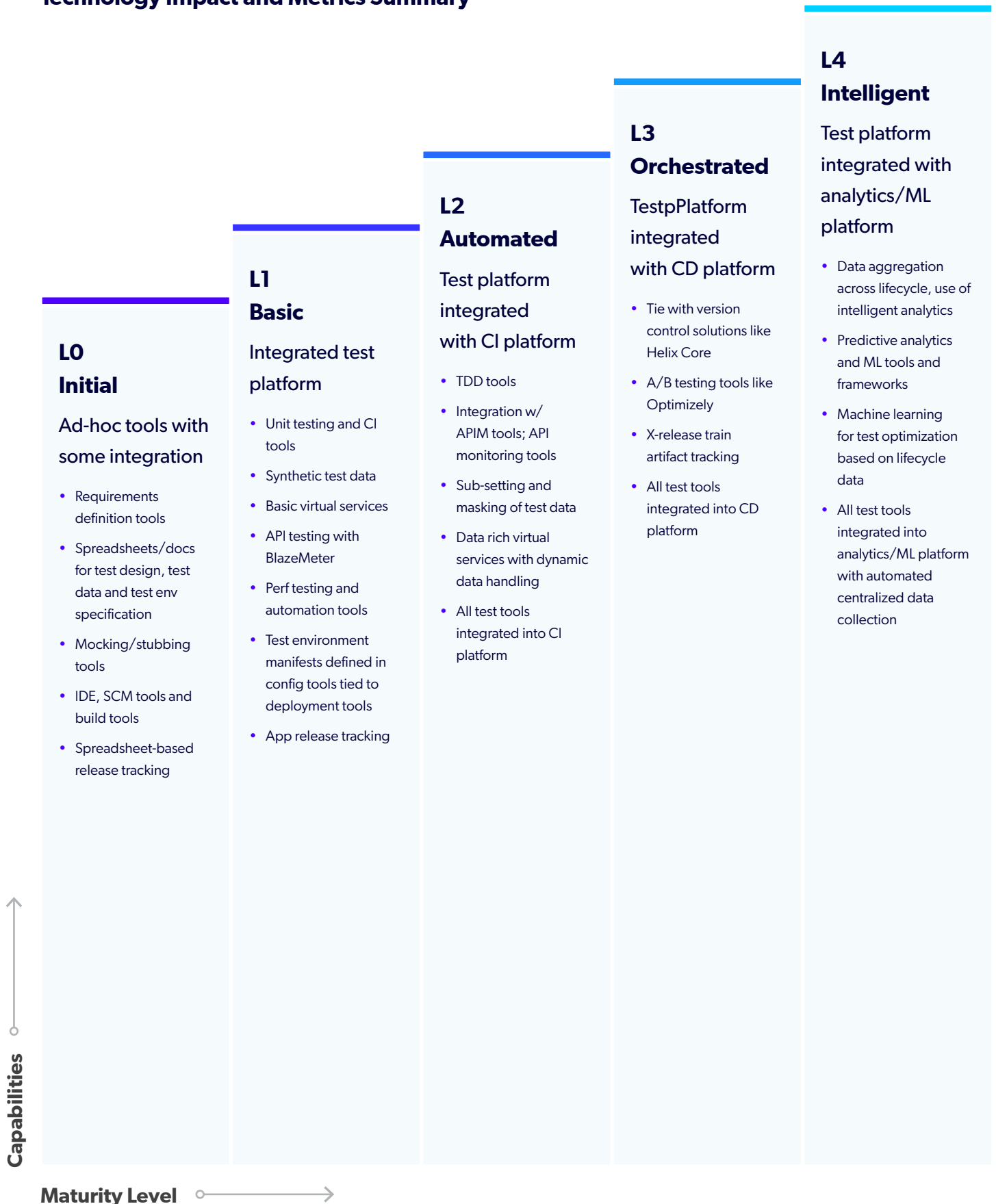
Process Impact – Orchestrated Phase

During the Orchestrated phase of process maturity there is a focus on CD integration. There will be a greater level of test automation (around 60 percent)—including full automated traceability, full test process/lifecycle automation and integration with CD processes. SDET/SRE skills will be needed at this level.

Process Impact – Intelligent Phase

In the final phase of process maturity, there will be a focus on predictive capabilities and optimization. Full testing CD lifecycle data collection, use of advanced analytics, data correlation, and ML, adaptive test optimization using ML, self-healing tests, and predictive testing capabilities are the hallmarks of this final phase. DET, test data science and analytics skills are incorporated into this phase of the continuous testing maturity model.

Technology Impact and Metrics Summary



Technology Impact – Initial Phase

When looking at continuous testing maturity model through a technology-based lens, the Initial phase is often defined by a use of ad-hoc tools with some (but minimal) integration. Spreadsheets are used for test design and release tracking, and mocking/stubbing tools will be utilized.

Technology Impact – Basic Phase

The Basic phase will include an integrated test platform, unit testing tools, some test automation tools, basic virtual services, API testing, CI tools, and app release tracking.

Technology Impact – Automated Phase

During the Automated phase, the test platform will be integrated with the CI platform. TDD tools (such as Gherkin and Cucumber) and automation tools (such as Selenium) are used, as are data-rich virtual services, API monitoring tools, as well as sub-setting and masking of test data.

Technology Impact – Orchestrated Phase

In the Orchestrated phase, the test platform is integrated with the CD platform. A/B testing tools are utilized, and artifact tracking is in place.

Technology Impact – Intelligent Phase

Finally, in the Intelligent phase of a technology maturity model, the test platform is fully integrated with an analytics and ML platform. CDD/CDI tools are used for data aggregation across the lifecycle, predictive analytics and ML tools and frameworks are in place, and machine learning is used for test optimization. Automated centralized data location is also in place.

Bottom Line

Whether you are in the Initial phase of your maturity model or you have integrated many of the processes of the latter stages of the maturity model, Perfecto is ahead of the curve when it comes to helping teams fully adopt a continuous testing maturity model.

From banking and finance, retail, healthcare, automotive, and telecom—to any other industry—Perfecto is your go-to partner for continuous testing.

Experience Perfecto in action today with our free trial or by watching one of our product demos.

[Start Trial ▶](#)

[Watch Demo ▶](#)

About Perforce

The best-run DevOps teams in the world choose [Perforce](#). Powered by advanced technology, including powerful AI that takes you from AI ambition to real results, the Perforce suite is purpose-built to handle complexity, maintain speed without compromise, and ensure end-to-end integrity across your DevOps toolchain. With a global footprint spanning more than 80 countries and including over 75% of the Fortune 100, Perforce is the trusted partner for innovation.

Harness the power of AI and accelerate your technology delivery without shortcuts. Build, scale, and innovate with Perforce—where efficiency meets intelligence.

About Perfecto

Perfecto powers exceptional digital experiences by combining the power of flexible test authoring, cross-platform execution, and AI-driven intelligent testing into one secure, cloud-based web and mobile testing platform.

Visit www.perfecto.io for a free trial, demo, or to learn more.

