

## Characteristics of a Successful TCOE

This white paper provides Chakkilam’s POV on how test organizations within the IT enterprise should approach as they build and operate centralized testing services. The goals of a TCoE organization is around how to do testing better, faster, cheaper. Based on Chakkilam’s experience in working with various clients and shaping their TCoE organizations, here are some testing best practices that might help organizations as they develop TCoEs within their IT organizations. The best practices focus on the areas of 1) People, 2) Process, 3) Governance, and 4) General Test Services.

### PEOPLE

A TCoE brings together a critical mass of experts that clients will view as constituting a real capability (as opposed to a scattering of individuals that are not able to provide high value testing services). The following are some best practices for establishing a TCoE from a “people” perspective:

- Manage the TCoE workforce to meet anticipated demand with dedicated professional testers
- Clearly define roles and expertise for members of this organization, including test leads, performance testers, environment support managers, test data managers, functional testers and defect managers
- Transform the organization to achieve best-in-industry testing capability and create test management discipline
- Cultivate and value testing skills and develop testing expertise across the enterprise, e.g. testing as a career path or test certification programs
- Provide an organizational home for resources who want to dedicate their careers in testing, and give them a career path for advancement
- Provide training and certification opportunities to testing personnel so that they may achieve best-in-industry testing capability;

### PROCESS

The following are some best practices for establishing a TCoE from a “process” perspective, the theme of which includes reusability so that processes do not have to be reinvented every time a TCoE is established:

- Acknowledge, tailor and integrate internal best test practices into TCoE test model
- Apply standard test methodology and leverage “best practices” to improve test quality and effectiveness



### About Chakkilam

Chakkilam Infotech is a 10 year old IT services company focused on software testing. We offer functional testing, test automation and performance testing services to software product companies and enterprise IT clients.

If you would like our sales team to get in touch with you, mail [info@chakkilaminc.com](mailto:info@chakkilaminc.com)

To know more about us visit [www.chakkilaminc.com](http://www.chakkilaminc.com)

- Establish standard test metric framework that supports fact-based decisions and is linked to IT and business values allowing for internal and external benchmarking
- Apply defect management standards across entire Solution Delivery Life Cycle (“SDLC”)
- Allow for iterative, more ‘agile’ testing concepts where applicable
- Create best-practice reference material, thereby preventing the outflow of corporate memory and enabling continuous improvement and re-visiting of that material

## GOVERNANCE

The following are some best practices for establishing a TCoE from a “governance” perspective:

- Operate TCoE as a service-oriented organization that yields value in excess of cost
- Establish and apply self-sustaining testing model with people, processes, tools, environments and services that can be deployed across the enterprise
- Provide a flexible engagement model for testing services across the enterprise that clearly defines services the testing organization will provide, along with roles and hand-offs
- Provide effective governance for end-to-end testing process
- Establish a Release Management process to enable structured releases of functionality into Production

## TEST SERVICES

The following are some best practices for establishing a TCoE from a “service offering” perspective:

- Balance quality vs. speed: Maximize “quick win” strategy (i.e., the “biggest bang for the buck”)
- Begin testing earlier in delivery lifecycle (e.g., validating testable requirements)
- Focus on testing high risk functionality (e.g., risk-based testing, eliminate non-value add testing and fix defects earlier in the delivery cycle)
- Integrate and apply both performance engineering and performance testing concepts

