T-Check: Low-Cost Approach to Technology Evaluation

Do you know whether the technology you plan to use will work in your context?

Do you want to verify claims made about technologies?

Do you think that it will be too costly or take too long to investigate technologies you want to use?

Today, most organizations are looking to become more agile and "do more with less" in order to stay competitive. As they look ahead, leading organizations rely on a proven approach to improved productivity and reduced cost—using the latest technology.

So, if your organization depends more and more on new or emerging technology, you'll want to know how you can be assured about the claims made for it. Will the technology work as promised and as expected in your organization's context? Will it allow you to easily modify your business processes, for example… or leverage your existing investments… or deliver platform independence?

T-Check: A Way to Test Technologies

The Carnegie Mellon® Software Engineering Institute (SEI) offers T-Check℠ investigations as an effective way to evaluate the appropriateness of a technology in your context.

T-Check investigations are experiments situated in a specific context, with the goal of providing a sanity check on a technology's claims.

T-Check experiments are ruthlessly efficient; they are simple investigations that provide insight into technologies without requiring a large investment.

The T-Check approach is like the classic scientific method in that it involves (as shown in Figure 1)

1. formulating hypotheses about the technology
2. examining these hypotheses against very specific criteria through experimentation
3. drawing conclusions from the test data about the usefulness of a technology in a certain context

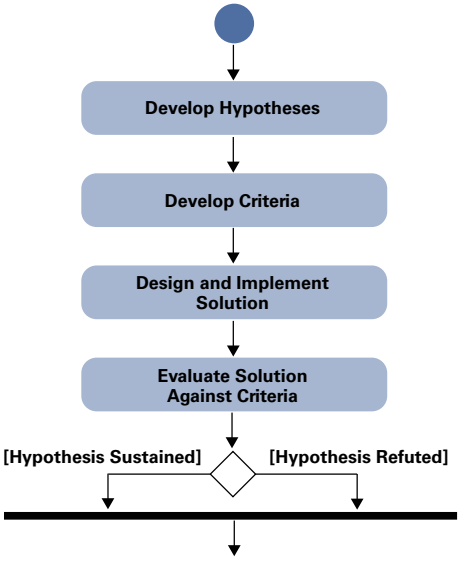

Figure 1: The T-Check Approach

You should consider the T-Check approach especially if you are planning for system-of-systems interoperability—through, for instance, an SOA infrastructure acquisition or a migration to an SOA environment, where many of the technologies and standards supporting SOA are still considered emergent.

Two Example T-Check Investigations

The System of Systems Engineering (SoSE) team at the SEI has performed T-Check experiments on some popular technologies for systems interoperability, including the following: Model-Driven Architecture and Web Services.

Model-Driven Architecture

For the specified conditions and through the T-Check approach, the hypothesis that the use of MDA reduces development time was partially refuted. Development time can increase greatly for the first application on which MDA is used, due to configuration and transformation modifications. The approach also refuted the hypothesis that the use of MDA frees the developer from understanding low-level details of the target platform and underlying infrastructure.

Web Services

For the conditions tested, the evaluation team determined that Web Services technology is fairly easy to implement. Likewise, it is simple to connect applications developed on different platforms using Web Services. The relative ease of implementing and using this technology is possible because Web Services elements are based on widely accepted standards that are supported by a large number of vendors. Nonetheless, the standards behind Web Services are still maturing. The T-Check approach also showed that too few public Web Services are available, and most of those available are poorly documented and of poor quality. (For more on this T-Check investigation, see the reverse side.)

Benefits of the T-Check Approach

- **The simplicity of experiments allows early insight into technologies without a huge investment.**
- **Clear hypotheses and criteria avoid time wasted "playing" with technologies.**
- **The additional findings are often greater than the direct results of the experiments.**
- **People conducting the experiments acquire early competence with the technology.**

T-Check: Low-Cost Approach to Technology Evaluation

T-Check Investigation of Web Services

Context for the Investigation: a military human resources system

Table 1: Hypotheses and Criteria

Hypothesis	Criteria
It is fairly easy for developers to connect applications developed for the same platform using Web Services.	• Documentation is available on how to implement and access Web Services in the selected platform. • Tools and libraries are available to implement Web Services in the selected platform. • Tools and libraries are available to generate code in the selected platform to access a Web-based service from the associated WSDL document that describes the service. • Two applications can connect using Web Services.
There are several public, easily discoverable, and high-quality Web Services that can be used in applications. (High-quality Web Services are those for which the interfaces are well documented and straightforward to use.)	• Developers are able to locate Web Services for use in their application by using public UDDI repositories or searching on the Internet. • The Web Services are well documented, and there is guidance on how to use them.
There are no problems relating to data types if Web Services are used to connect applications on different platforms (e.g., J2EE and .NET)	• The two applications can exchange complex, date, and floating point data types with no data inconsistencies between the two platforms. • The exchange can be done using default mechanisms provided with the Web Services tools and libraries.

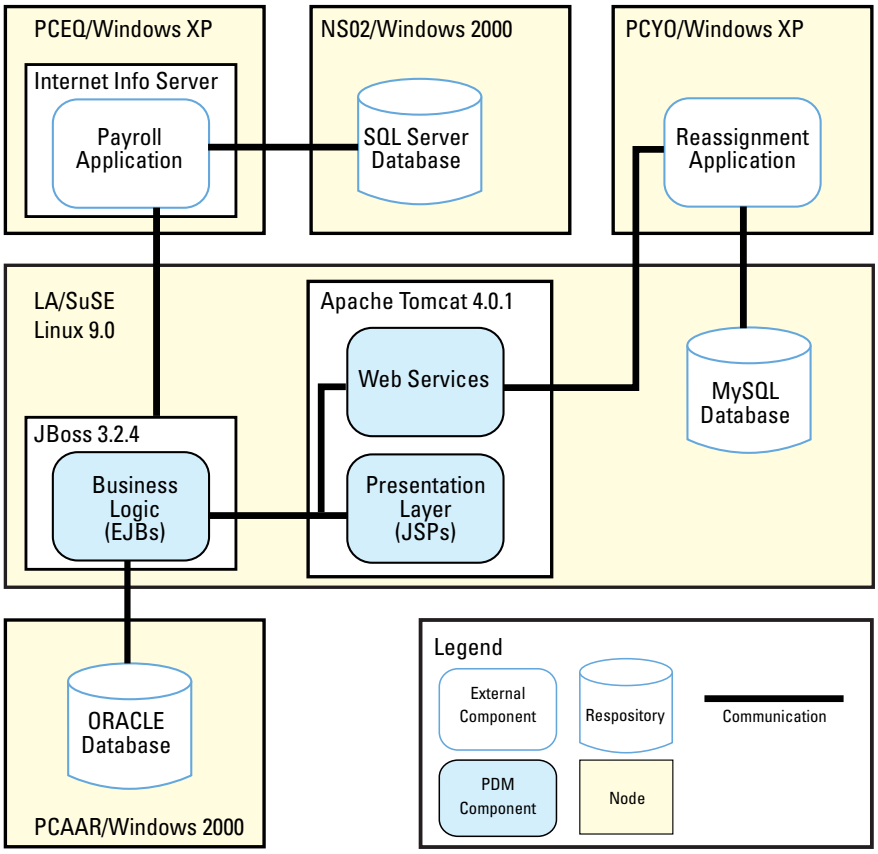

Figure 2: Deployment View of Model Solutions for T-Check Investigation of Web Services

Related Website

www.sei.cmu.edu/interoperability/casestudies/techeval/

For General information

For general information about the SEI and its products and services, contact

Customer Relations
P: 412–268–5800
F: 412–268–5758
www.sei.cmu.edu
Software Engineering Institute
4500 Fifth Avenue
Pittsburgh, PA 15213-2612

T-Check Reports

Model Problems in Technologies for Interoperability: Web Services (CMU/SEI-2006-TN-021).
http://www.sei.cmu.edu/library/abstracts/reports/06tn021.cfm

Model Problems in Technologies for Interoperability: OWL Web Ontology Language for Services (OWL-S) (CMU/SEI-2006-TN-018).
http://www.sei.cmu.edu/library/abstracts/reports/06tn018.cfm

Model Problems in Technologies for Interoperability: Model-Driven Architecture (CMU/SEI-2005-TN-022).
http://www.sei.cmu.edu/library/abstracts/reports/05tn022.cfm

T-Check in Technologies for Interoperability: Business Process Management in a Web Services Context (CMU/SEI-2008-TN-005).
http://www.sei.cmu.edu/library/abstracts/reports/08tn005.cfm

T-Check in Technologies for Interoperability: Web Services and Security—Single Sign-On (CMU/SEI-2008-TN-026).
http://www.sei.cmu.edu/library/abstracts/reports/08tn026.cfm

T-Check for Technologies for Interoperability: Open Grid Services Architecture (OGSA)—Part 1 (CMU/SEI-2007-TN-016).
http://www.sei.cmu.edu/library/abstracts/reports/07tn016.cfm

A Process for Context-Based Technology Evaluation (CMU/SEI-2005-TN-025).
http://www.sei.cmu.edu/library/abstracts/reports/05tn025.cfm

03/25/2009