Application Architecture

Committed Partner. Creating Results.

Table of Contents

I. Can Your Application Architecture Accomodate your Changing Business Needs?

Continuous change that results in dynamic environments poses a serious business challenge. To manage the cost of change, organizations need to be agile. The key to agility is having adaptable business systems. Building and maintaining adaptable business systems requires agile development processes. Large-scale systems, in particular, also need to be based on a flexible architecture that can accommodate change. Two fundamentals for a flexible architecture are component based development and technology neutral models.

The old cliché about change being the only constant holds true for technology. In today's world, rather than expecting stability between periods of change, we recognize that change is continuous. This can be a great detriment to business because many times a company's new technology is outdated before it realizes its return on investment.

Not only has technology changed, but the ways we interact with and use technology to achieve business goals has also changed. Yesterday, companies had to install expensive Value Added Networks (VANs) in order to trade with their business partners. Today, the Internet allows business partners to trade and exchange data in Business-to-Business (B2B) configurations. Yesterday, companies were geographically limited to their retail outlets. Today, companies use the Internet to offer products and services through Business-to-Consumer (B2C) e-business channels that reach customers all over the world. Within an organization, Intranets allow streamlining of organizational business processes and team collaboration, reducing costs and creating economies of scale. Tomorrow, wireless devices like PDAs will again change the ways we send and receive information. Users will be able to interact with enterprise systems from anywhere in the field, allowing more flexibility and mobility, and resulting in more change.

For companies to compete successfully in the future they will need to embrace change by building adaptable business systems using agile development processes and basing them on flexible architectures.

Key Issues

How can companies ensure that their application architecture is flexible enough to adapt to continuously changing business needs in a dynamic IT environment?

Key Facts

- One of the greatest problems in the development of modern software systems is planning for change: open systems must be flexible in that they must be easy to adapt to new and changing requirements.
- Flexibility is especially critical to e-business because of the need to react quickly to competitive innovations and partnership opportunities.
- Application architecture's core business benefits are flexibility and investment protection. In addition, it can reduce development costs and increase application quality.
- Application architectures also need to enhance stability, reduce complexity, and realize cost-savings through software reuse or adaptation.
- A good way of dealing with changing requirements is to build systems out of reusable components that conform to a "plug-in architecture". In this way, functionality can be changed or extended, either by substituting components, or plugging in new ones.

Source: *Giga Information Group*

II. Agile Organizations Embrace Change

An organization's competitiveness is determined by the speed at which it adapts to change. Agile organizations capitalize on changes in the market by quickly adapting to them, resulting in a definite competitive edge. They do this by embracing the notion of continuous change, setting in place a mindset that allows for the rapid redefinition of business processes, reorganization of staff, and realignment of technology in support of new business goals. At their core, agile organizations are streamlined in their organization structure, processes and technology, with all three acting in alignment with each other, as well as in alignment with their strategic goals.

The key to embracing change and developing into a technically agile enterprise lies in the development of *adaptable business systems* – systems that can change quickly in response to changes in either the processes or the organization they support.

III. Agile Development Processes Help Flatten the Cost-of-Change *Curve* When Developing Adapatable Business Systems

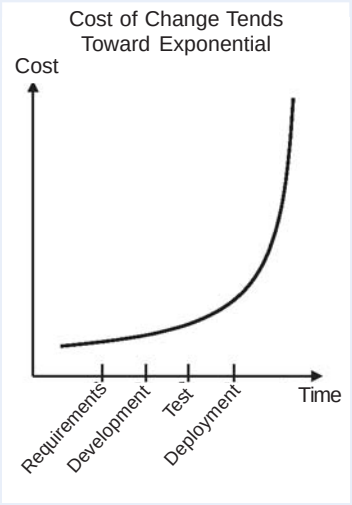
Much excitement (and controversy) prevails in the industry today about agile application development processes such as eXtreme Programming (XP), SCRUM and others. These methodologies address application development processes, and provide a means to develop quality application code that is eminently adaptable to change. Agile development processes help greatly in developing adaptable business systems that help flatten the cost-of-change curve. However, the development of adaptable large-scale distributed systems requires flexible application architectures in addition to agile development processes.

There are two additional fundamentals that architectures need to leverage in order to be flexible: technology neutral architecture models and component based architectures.

Flattening the Cost of Change Curve

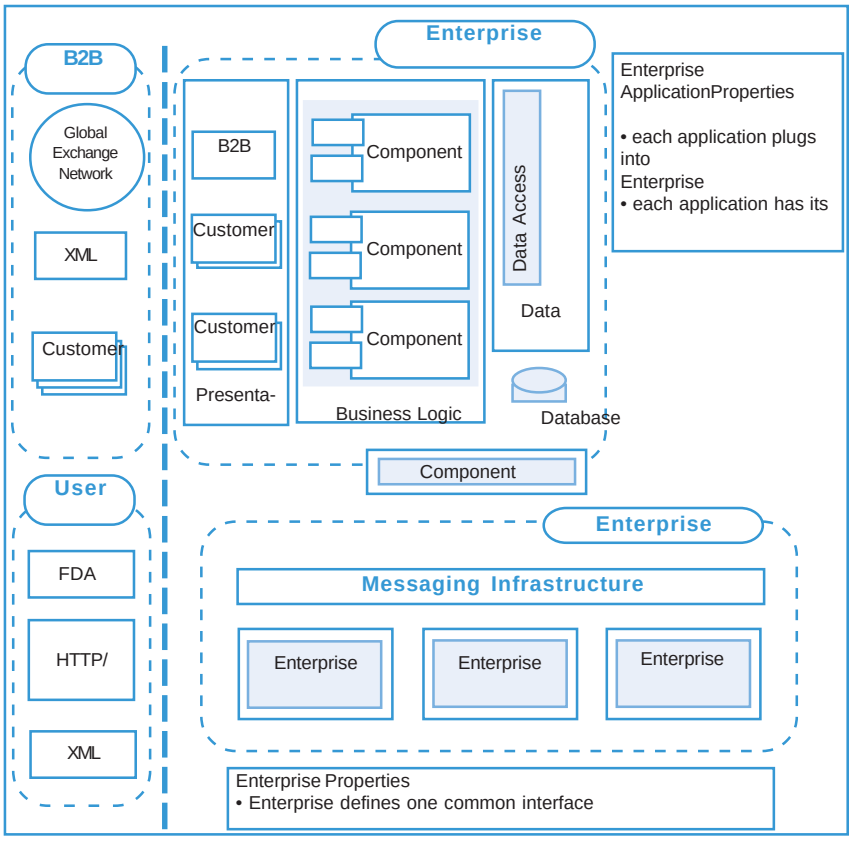
Kent Beck
author of the eXtreme Programming (XP) process posits:

"Under certain circumstances, the exponential rise in the cost of changing software over time can be flattened. If we can flatten the curve, old assumptions about the best way to develop software no longer hold."



IV. Technology Neutral Architecture Models Help Avoid Building "Instant-Legacy" Systems That Cannot Adapt In The Future

A technology neutral architecture model is a generalized architectural model that is based on a higher-level and more abstract view of business models. Such an architectural model is developed to show large-grained subsystems and their interaction with the rest of the enterprise, as well as their interaction with new and future external systems. After such an architectural model is developed and analyzed, it can then be implemented using the best technology available.



The above technology neutral architecture model helps to identify all the components that will be required to provide the business services needed, and to identify the dependencies between them. If a business process changes, the architecture model enables handling the change by identifying unwanted interfaces that need to be removed, and new interfaces that need to be developed. This approach avoids designing "instant legacy" systems that cannot adapt in the future.

Key Issues

How can companies ensure that their application architecture is flexible enough to adapt to continuously changing business needs in a dynamic IT environment?

Key Facts

- One of the greatest problems in the development of modern software systems is planning for change: open systems must be flexible in that they must be easy to adapt to new and changing requirements.
- Flexibility is especially critical to e-business because of the need to react quickly to competitive innovations and partnership opportunities.
- Application architecture's core business benefits are flexibility and investment protection. In addition, it can reduce development costs and increase application quality.
- Application architectures also need to enhance stability, reduce complexity, and realize cost-savings through software reuse or adaptation.
- A good way of dealing with changing requirements is to build systems out of reusable components that conform to a "plug-in architecture". In this way, functionality can be changed or extended, either by substituting components, or plugging in new ones.

Source: *Giga Information Group*

V. Component Based Architectures Support Changes Across Applications and Technologies

A component-based architecture provides a framework for designing and packaging applications into layers, components and modules. This provides a foundation that is consistent across applications, and relatively stable across changes to the technologies underneath, as well as, to the application portfolio above.

A *component* is a software unit that encapsulates both data and code, and provides a well-specified set of publicly available services, known as its *interface*. Because the way a component is implemented can be changed without affecting its public interface, new components can be "plugged-in" or old ones extended to provide a level of flexibility in dealing with changing requirements. Component based architectures that employ loose coupling[1] increase reusability and also make the architecture easier to understand, maintain, and change.

VI. Conclusion

In addition to agile development processes, technology neutral models and component-based architectures are fundamental to the development of a flexible architecture that supports adaptable large-scale business systems.

To manage the cost of change in the face of continuous change, organizations need to:

- Embrace change in order to be agile and competitive by developing adaptable business systems.
- Implement agile development processes to support the development of adaptable business systems and flatten the cost of change curve.
- Base the development of adaptable large-scale business systems on a flexible architecture.
- Leverage technology neutral models and component-based development to construct a flexible architecture.

VII. References

1. *Realizing e-Business with Components*, Paul Allen, Pearson Education Limited, 2001.
2. *Large-Scale Component-Based Development*, Alan W. Brown, Prentice Hall, 2000.
3. *eXtreme Programming Explained*, Kent Beck, Addison Wesley, 2000.
4. *Object Technology – A Manager's Guide,* David A. Taylor, Addison Wesley, 1998.

1 Coupling *refers to the level of interdependence among the components of a system. Tightly coupled components are highly interdependent, and result in an architecture that is brittle and resistant to change. Conversely, loosely coupled components are less interdependent and allow the flexibility required to support change.*

Sanjiv Augustine is a Technical Director with CC Pace, a technology-based business solutions consulting firm headquartered in Fairfax, Virginia. George Kosmides is co-founder and vice-president of Noospherics Technologies, Inc, a partner firm of CC Pace. You may contact the authors by e-mail:

sanjiv.augustine@ccpace.com or gkosmides@noospherics.com or reach them by mail or telephone at:

CC Pace
4100 Monument Corner Drive
Suite 400
Fairfax, VA 22030
(703) 631-6600

For more information, visit our website at http://www.ccpace.com.