



Considerations for Component-Based Development

Robert Armstrong, Gillian Adens

Tassc Limited
www.tassc-solutions.com

First Published: April 1999
Last Updated: January 2008

Copyright 1999-2010, Tassc Limited. All Rights Reserved.

Tassc is an independent software company with considerable experience in helping major organisations establish and succeed with component-based approaches to the development of business solutions.

In particular, we have real project experience in the development of component architectures to support and drive both business change and software development. Our approach is non-proprietary, method independent and based on best industry practice.

Over the years we have gained many valuable insights. In this paper, we examine and discuss key issues facing major companies as they assess their current information systems and prepare for the adoption of a new approach to information technology.

Can we protect our investment by integrating legacy applications?

*develop
component
wrappers for
existing
functionality*

Utilise legacy systems wherever practical to do so. Useful and cohesive components in existing systems are identified by matching system functionality against the business architecture. The functionality of the system is then wrapped to provide well-defined components with service-based interfaces. In this way the investment in existing systems can be leveraged to provide significant productivity gains.

*align business
and technical
integration
strategies*

For integration of existing systems with newer technologies, a comprehensive strategy covering both business and technical integration issues will be required. Business oriented components can be identified which are implemented in existing systems. A robust and efficient technology strategy will be required to ensure that new applications can, and do, access and use these components wherever possible. The technical strategy will also cover integration where components span multiple existing systems.

How do we develop a coherent strategy for migration?

*encapsulate
commonality in
components*

Development costs and ongoing maintenance costs may be higher than necessary. There may be opportunities to reduce these costs by identifying common functionality to implement as components for use across multiple systems. A business architecture is used to identify commonality and to define a series of cohesive components that encapsulate this commonality.

*avoid duplication
of effort and
facilitate
migration*

With well-defined components, maintenance is isolated in a single location rather than spread over multiple systems. This avoids duplication of effort and provides for a controlled, lower risk migration option. Provided the published component interfaces are adhered to, components can be individually and independently replaced by new systems, as and when it is considered appropriate and with little or no impact on existing applications.

Can we afford *not* to pursue new technology?

*understand the
long term
business and
technology
drivers*

The organisation will need to determine the impact of not adopting new technologies and approaches. It will need to assess its ability to exploit technology for competitive advantage. The agility of the business should be understood. Competitor business activity and their exploitation of technology will influence demands for systems capability. Trends in IT, rather than fads, are important drivers. Labour market trends are equally important. Support policies from existing suppliers for older technology may also force technical change.

Is the business moving towards flexible product features?

*develop modular
components for
maximum
flexibility*

Many industries are implementing more flexible product development strategies by defining low-level product feature sets as product modules. The product development function or, in some industries, the customer can then make a choice amongst the modules to create a personalised product. This business driver is the motivation for creating a series of software components that support the product modules and can easily be integrated or extended to support rapid product development in the business.

How will new technology reduce costs?

*develop a
software
architecture*

The development of a software architecture, (the large-scale identification, structure and organisation of software components and their interfaces) provides enterprise wide system integrity and identifies opportunities for strategic reuse.

*plan and design
for reuse*

Over time, aim to establish a library of reusable components, either developed in-house or bought-in. The process of developing new applications can then be more of an integration effort rather than always resorting to developing everything from scratch.

*good design
reduces costs*

Well-designed components encapsulate both data and functionality, and provide a well-defined published interface. These software elements are inherently modular (with high cohesion and low coupling) and once developed can be extensively reused. Reusing software that has been previously analysed, designed, constructed and tested, allows more rapid application development and improves quality. Better quality applications have fewer defects and require less maintenance.

How do we deliver systems to the timetable required by the business?

deliver high quality with greater speed and flexibility

Ensure that the business understands that the benefits of component-based development are not instant. There is certainly an initial up-front investment. However, thereafter it should be possible to deliver business applications of high quality with greater speed and flexibility.

take application development off the critical path

The window of opportunity to gain an advantage over competitors is typically very small. Information systems support is often critical to new product and service initiatives. The time to market is often constrained by software development. Business focused components should reduce development time and help to take application development off the critical path.

How do we prioritise, plan and schedule business projects?

determine business priorities and dependencies

Assign a high priority to projects that bring the greatest business benefit (through increased efficiency, lower cost, uniqueness of service, competitive advantage, etc.). Identify dependencies between projects and any common services that need to be shared. Initiate projects that provide common services early (infrastructure projects), followed by projects (in priority order) which use these services (application projects).

How much of an initial investment needs to be made?

understand commitment and skills

Assess the existing company culture, particularly staff attitudes towards new technology. Examine the development organisation structure. Assess commitment of senior management. Understand skill levels both of development staff and management. Be clear on the motivation behind change. Always aim for evolution, rather than revolution.

work on people issues in parallel with technical change

Typically a small team of capable analysts and designers can be formed, with guidance from an experienced mentor, to initiate architectural and process change. Working on a pilot project, they should start to deliver tangible business benefit within months. In parallel a small management level team will be required to work on organisation and cultural issues.

How do we negotiate the steep learning curve?

establish corporate processes and a corporate design authority

There is a learning curve that must be tackled if component-based development is to be adopted. Conceptually there should be few significant difficulties since many of the principles of object technology and component development are the same as traditional development. The difference is that current tools and techniques give greater support to best practice. However, component development implies significant interrelationships between different developers. In order to control this, establish corporate processes that encourage a positive attitude to component reuse. A level of discipline is required and corporate design authority must be respected. This is the job for the company's most senior and respected software engineers.

invest in skills development at all levels

Invest in management and staff training. People are often highly motivated through learning new skills. Demonstrate commitment from the business by communicating business objectives and empowering staff. Provide adequate support from experts with a proven track record of similar assignments.

What is the best process for component-based development?

*use an
architecture-
centric, iterative
approach*

An architecture-centric, iterative process based on business requirements offers the best route to success while minimising risk. First, establish an outline architecture in order to impose structure and encourage system integrity. Refine those areas of the architecture that are required to support application projects that address high-priority business objectives. Develop application projects in a phased manner, where an application is evolving over time with a series of planned releases made into the business.

Do we need to run a pilot project? And how do we identify candidates?

*clearly identify
pilot project
objectives*

Run a pilot project where little or no experience with new technology exists in-house. Choose a non-mission-critical application, but one that is not too trivial so that valuable practical experience can be gained. Ensure that the objectives for the pilot are clearly identified and fully understood. Pilots projects are most successful if the development team is small, multi-skilled and has access to an experienced project mentor.

Do we need to change the structure of our development organisation?

*develop a
supportive
organisation and
culture*

Ideally development should be organised around multiple teams, each specialising in a particular domain (architecture, common services, user interface, database, business applications). Each team should be aware of their contribution to business objectives, and that they are part of a culture that encourages component reuse and development of components *for* reuse. The organisation is partitioned into teams that supply reusable components and teams that consume component services to construct business applications.

*attitudes to
change are
absolutely
fundamental to
success*

The cultural changes are always the most difficult. Essentially we are taking people outside their 'comfort zone' and asking them to learn new skills and adopt new ways of working. Attitudes to change are absolutely fundamental to success. Nonetheless, senior management can make a significant difference by communicating the advantages and benefits of the new strategy. Software engineers react much more positively if they feel involved and empowered by a component-based approach, rather than coerced and restricted by it.