

Application Transformation

Needs Assessment and Options Analysis

Crisis: For years you have been enjoying the low cost of ownership of your Progress ABL application, investing modestly in functional modifications. You might be aware that there were many new capabilities in the language and database which were not being used, but you were comfortable and familiar with what you had and saw no need for change. If you were familiar with some of the new architectural models being discussed for Progress applications, you thought they applied to people writing new applications and had no immediate relevance to someone with a million lines of existing code. Bottom line, your application is finely tuned to your business and the work was getting done, so why change? Then, a crisis hit:

- New management or your customers demand that you replace the existing character user interface;
- You need to web-enable parts of the application for remote users, customers, or vendors;
- There is a demand or critical opportunity for supply chain integration from customers or vendors;
- There is a core shift in your business which requires a logic change in the software and you realize that you don't know where the change needs to be made and that the logic is duplicated in many places; or
- Your expanding company is no longer functioning well with a single large centralized solution.

But, whatever the trigger, suddenly you recognize that the code needs to change significantly, more significantly than it has changed for years and years. And, not only don't you know where you are going to find the resources, but you realize that the very architecture of the application is likely to have to change. Worse yet, you aren't even sure what the target architecture should be or, even if you think you do, you have no experience creating applications with that architecture and no idea how to change what you have into what you need. I.e., you know that the software needs to be transformed or re-invented somehow, but not exactly what the target should be or how to get there.



Business Drivers: While the specific stimulus that provokes a recognition of the need for transformation varies from company to company, it is likely that there are many other issues lurking below the surface for any company with a legacy application. I.e., it is common for companies to have many problems that have simply not become critical enough to precipitate a crises, but which are nevertheless real issues with the application. A company that decides that they need to web enable part of the application for remote users may also have some growing unhappiness with the general user interface and might have opportunities for integration which have

simply not become important enough to be the factor that has tipped the scales. A company which decides that they need to service enable an application for supply chain integration probably also has issues about the user interface and maintainability issues. Most companies have issues about poorly documented and poorly structured software which simply aren't critical until more major changes are required.

Thus, in any company there is a mix of primary business drivers – those issues that precipitated the crises – and secondary business drivers – other issues that might have eventually produced a crisis, but are not yet

perceived of as being critical problems. These secondary drivers can be the same issues which are critical for another company, e.g., user interface for company whose critical problem is remote access, or they can come from general characteristics of legacy applications, factors which may simply be accepted as “the way things work”, but which are, in fact, significant business issues or opportunities.

Among the kinds of secondary business drivers which are characteristic of legacy applications, some examples are:

- Maintenance may have become an increasing problem because the code is largely undocumented and very heavily patched;
- A once acceptable ease and speed of modification may no longer be adequate as the pace of business change has increased; or
- A centralized architecture may no longer be optimum as the company spreads geographically.

Some of these factors result in real out-of-pocket costs, which have the potential for being reduced, while others represent lost opportunities for improved revenue or profit. Speed and ease of making changes to an application is an increasingly important opportunity for many businesses because of dramatically shortened business cycles in the modern world.

So, the first thing a company should do when they decide they need transformation is to look at the overall application and its fit to the business – where it is working well and where it isn't; where it could be improved; what cost savings or benefits or new opportunities might arise; and what are the pros and cons of various alternative approaches and solutions. It is important to make this assessment up front for a number of reasons:

- First, there may be substantial synergies between changes for one purpose and those for another, so that there would be substantial cost savings from a combined solution over two separate projects;
- Second, while the immediate trigger driving transformation may seem important, an honest analysis may reveal that it is a less substantial business driver than was assumed and there are other factors which are more immediately important.
- Third, there are many alternative strategies to transformation and they have widely different costs, time frames, and impacts, so these need to be analyzed in relationship to the real business benefits which each would produce. A path that seems economical in the short run may not sufficiently address pending issues and end up being more expensive in the long run when additional changes are required.



Needs Assessment: In order to provide a baseline for evaluation, Computing Integrity will work on site evaluating the existing application and its relationship to the business. This evaluation covers an assessment of the software itself in terms of current architecture and potential for change, as well as the resources available in existing staff for making changes. The assessment also includes interviews with key personnel to determine business needs and opportunities and with users to assess the current usability

of the software and its potential for impact. Our intent is to determine not only where the business is today, but where it wants and needs to be in the future. We have extensive experience with many types of businesses and have often consulted with our clients on business issues in addition to their technical solution, so we are able to learn the essential features of a new business with only a limited exposure. Following the on-site visit, the results of the assessment will be prepared and presented to the company for validation and can be used as the basis for discussion of alternative solutions.

Transformation: There are many alternatives available for transforming legacy applications so that they better fit modern business needs. Two of these are characterized by modest short term investment:

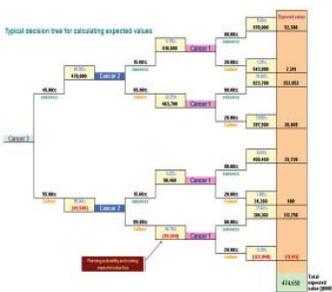


- **Spot Changes:** In some cases, the application as a whole may not need transformation, at least not immediately, and one can resort to a sharply focused solution. A spot change will not modernize the application or gain the agility benefits of a modern architecture, but can be a pragmatic, short term solution to an immediate problem. Spot changes are not appropriate for general architectural shifts, but can be used when there is an isolated function needing new capabilities. However, such localized modifications only perpetuate any architectural issues which exist and may even make them worse.
- **Stepwise Transformation:** Alternatively, one can consider one of several types of stepwise transformation in which a general architectural goal is determined and then applied on one problem area at a time, typically beginning with those having the highest ROI benefits, but continuing systematically until much or all of the application has been transformed over a number years. One variation on this approach might involve secondary transformations of selected subsystems when a significant number of the components have received their initial transformation, e.g., stepwise transformation to a Service-Oriented Architecture and then creating a new user interface technology for a subsystem when it is largely converted to services.

There are two primary strategies for a more complete, short term transformation:

- **Single Focus Complete Transformation:** In this approach, some key element needing transformation is identified and the application undergoes a major rewrite to transform this one area, possibly with some secondary attention to other areas. The result can be a substantially altered application, e.g., moving from a character to graphical interface, but the system has not been fully modernized or re-architected since it is basically a component for component transformation. In full modernization, the components of which the application are composed often change significantly from that in legacy applications, not just in internal implementation, but in overall packaging.
- **Full, Model-based Transformation:** Alternatively, there is full model-based transformation in which one extracts business rules from the existing code, supplemented with traditional Object-Oriented Analysis and Design (OOAD) techniques, to create a full OOAD model of the application and then one generates a complete new application from this model using Model-Driven Architecture (MDA) techniques. The result of this transformation is fully modern in all respects and provides a mature, top quality application which can continue to evolve rapidly based on the model driven approach. While it is the most extensive change of any approach, it has the potential for significant efficiencies and leveraging through the use of tools and thus potentially can take less effort and time than a single focus transformation.

There are many variations and shadings on each of these basic types, but this classification helps to identify four primary strategies which can be used in responding to the need for substantial change in an application.



Options Analysis: Following the Needs Assessment, we will evaluate the various transformation alternatives in relationship to the business drivers, costs, benefits, potential revenues, effort, risk and other considerations in order to provide an overview of the options available to the company for improving their application. This analysis should provide the company with a sound basis for determining which alternatives need deeper evaluation and specification and which can be left aside for one reason or another. While detailed evaluation and planning will be required to develop a full transformation plan, this report should

provide the company with the basis for initial budgeting and making strategic decisions about the steps needed to adequately respond to the perceived crisis and any other issues which are identified in the course of the study.

Overview of Offering: This product provides for the assessment of:

- Existing software architecture, quality, and soundness;
- Capacity and characteristics of the existing software team;
- Business drivers related to both the focus of the immediate crisis and other issues known to the company or discovered in the course of the assessment.
- Growth and functionality changes required to achieve goals for future business requirements; and
- Appropriateness of alternative transformation strategies for the company's needs.

Initial deliverables will include:

- Report on the current overall state of the software, existing issues and potential for change;
- Report on business drivers and opportunities; and
- Analysis of the relative merit of the alternative options for transformation and change'

Anticipated billable services and estimated level of effort:

- Review, interviewing, and data gathering; 4 days (onsite)
- Documentation and analysis of application status and business drivers; 2 days (offsite)
- Analysis and report on transformation options; 1 day (offsite)

Services are billed at prevailing standard rates. Travel and expenses for on-site work are extra.



COMPUTING INTEGRITY
INCORPORATED

60 Belvedere Avenue
Point Richmond, CA
94801-4023

Voice: 510-233-5400
E-Mail: thomas@cintegrity.com
Web: www.cintegrity.com