Lean Application Development
A CAI White Paper by Nicholas Spanos

OVERVIEW

The evolution of the Information Technology industry is following a path that is quite similar to the evolution of the manufacturing industry. Just like the Manufacturing industry of the 1980's, IT depends on highly skilled craftsmen to custom build applications based on a custom specification that is unique for each customer.

Every year there are studies published that show that IT development and integration projects continue to be challenged and have a high failure rate. There are many opinions regarding the cause of these high failure rates. IT project teams tend to blame the business for failing to provide detailed requirements and the business blames IT for failing to understand the requirements and provide technology leadership.

While these issues are real and need to be addressed, they are exacerbated by IT's approach to Application Development which does not effectively manage creativity and does not leverage re-usability beyond basic objects. The Agile framework was designed to provide a structure to this type of iterative development but significant issues remain that impact cost, quality, schedule, and customer satisfaction. The Cloud approach seeks to eliminate the problem by suggesting that organizations accept standard functionality to support their business processes. Unfortunately, the business commonly resists changes to business processes and there are numerous security challenges that have yet to be resolved so we still need to address Application Development challenges.

UNDERSTANDING THE PROBLEM

ANALOGY: MANUFACTURING REVOLUTION

In the 1800's, the manufacturing industry was faced with similar problems. They were heavily dependent on skilled craftsmen to make unique products. Henry Ford standardized production and assembled cars from standard components instead of the slow process of crafting an automobile. His approach to a manufacturing continues to evolve today with concepts like Lean Manufacturing.

1. Prior to the 20th century, manufacturing involved a fabrication process employing highly skilled craftsman. Quality was directly linked to the professionalism and skill of the individual craftsman. The resulting products were individual creations that may have been similar but were rarely identical.

CAI
Computer Aid, Inc. ®
World Leader in IT
Metrics and Productivity

2. Most manufacturing companies followed the lead of Henry Ford and transformed from a fabrication process employing highly skilled craftsmen to an assembly line using lower skilled workers to assemble standard components. Results: Greater throughput, lower costs, standardized quality.

3. The creative abilities of skilled craftsman were no longer necessary when building products. Creative activities occurred during the design of the components and the development of the assembly processes. The effort required to assemble the final product was easier to control and dramatically increased the quantity, quality, and consistency while reducing cost.

4. Quality problems persisted because of issues with component design and the assembly processes. The second revolution in manufacturing came with the increased use of automation to assemble components, better metrics, and better quality control.

5. Results: The price of most manufactured goods continues to decline (relative to inflation) while the quality and features continue to increase.

THE CURRENT STATE OF APPLICATION DEVELOPMENT:
This section will discuss how challenges with IT management

1. Most IT organizations develop Software in response to a unique set of requirements. They employ skilled craftsmen who utilize their creative skills to develop software that is customized for the stated requirements. This approach is similar to the use of craftsman by the manufacturing industry prior to the 20th century (with similar results). Each product is unique and quality and efficiency are the responsibility of the individual craftsman and difficult to measure.

2. The quality and efficiency of the design and the final product is the direct result of the creativity and skill level of the individual developers. Functionality is based on the unique requirements as defined by the requester.

3. The effort (and the cost) for designing and developing these unique "one off" applications is significant. As a result, new applications are rarely authorized and we continue to operate and maintain older legacy applications. Support costs are also higher because the resulting applications require special knowledge to operate and support because they are all uniquely different in their basic design.

4. Collecting metrics is difficult. Creative processes do not automatically generate metrics. Metrics generation requires additional effort that is typically resisted. Even if metrics are collected, they are useful in defining and measuring quality but the existence of metrics to not ensure quality.

5. Repeatable processes are marginally useful in defining and managing creative fabrication activities. They are very useful in managing repeatable assembly activities.

6. Development tools change frequently. The decision to utilize a new tool is very often made by a developer without serious consideration for the available knowledge base and future support challenges

7. Current development tools provide the capability to leverage the enhanced infrastructure. Most of these new Object-Oriented languages are based on low-level rudimentary languages like C++ and Basic. They facilitate code re-usability at the object level (e.g. drop down boxes) but they hinder re-usability at the functional level. In the past, we could develop an online interactive program that updated a database and clone this program to develop other similar functions. This level of re-usability is much more difficult with Object Oriented languages.

8. These new Object Oriented Languages are more difficult to learn than their predecessors. Inexperienced developers are more likely to have learned the newer languages but they do not have the design skills required to develop new applications.

IT organizations do not train their experienced developers on newer technologies. As a result, the average skill level available for newer technologies is not adequate for developing complex applications.

THE OBVIOUS SOLUTION: ORGANIZE DEVELOPMENT TEAMS BY TECHNOLOGY AREA

1. Choose the desired development environment(s) and technology by assessing the benefits and limitations of the available options. The decision should also consider the availability of knowledgeable staff and the effort required to learn the technology.

2. Create core teams that specialize in specific development environments and tools. The staff must be pre-trained and include a variety of skill levels to allow for career growth opportunities.

3. Assign a system architect to each development environment. The architect should have strong design, development, and support experience with the environment.

4. Assign an experienced manager with process development and enforcement experience as well as strong organizational skills.

5. Create repeatable estimating processes that consider the unique requirements of each technology area.

6. Maintain team cohesion. At the end of a project, team members are dispersed to other assignments. It is much more effective to retain team unity and assign them to the next project as a unit.

7. Define Formal Roles and Responsibilities. The roles and responsibilities should consider the unique requirements of each development environment and also be linked to position descriptions and career development paths. Assignments should consider prior experience as well as the development plan of associates to facilitate career development that will be valuable in future projects. Sometimes, this choice will have a short term negative impact on

profitability that should be considered as an investment to increase profitability of future projects.

8. Define standard documentation formats. Design specifications facilitate development but they provide limited benefits once the system is implemented in production. Standard Application documentation is required to assist with the operation and support of the application.

9. Support considerations should receive a high priority. Design decisions can have significant impacts on the supportability and reliability of applications. A high level of manual intervention combined with inadequate data validation and automated error trapping account for a large number of recurring problems. These decisions may reduce development cost and expedite development. Unfortunately, the increased support costs and risks will eclipse any development savings.

THE NOT-SO-OBVIOUS SOLUTION: LEAN APPLICATION DEVELOPMENT EMPLOYING PLANNED RE-USABILITY TO MANAGING CREATIVITY

ISSUE: The creation of teams that specialize in a given technology is a common practice. These teams continue to fail because they still depend heavily on creative efforts of the individual team members.

"Object Oriented" languages claim to provide re-usability. Unfortunately, their re-usable components are low-level objects. They provide re-usable functions for creating a drop-down list but do not provide a re-usable function for updating information about people.

SOLUTION: We need to be more proactive in developing and utilizing re-usable components. We also need to manage and control creative activities that are difficult to estimate and can have significant impacts on the schedule, quality, risk, and cost of a project. The following recommendations apply to each technology area:

1. Create re-usable specifications and functions. Most systems maintain data about people, places, things, and processes. When designing screens and databases to maintain this type of information, choose design options that provide the greatest potential for re-use in future applications.

Most applications track people and organizations. Unfortunately, these functions were designed to be unique to each application and are not re-usable across applications. Re-usability should be a high priority design consideration.

2. Use bench resources and trainees to create/enhance re-usable modules. Gradually, an inventory of high quality modules will be available for each technology area that will facilitate future development.

3. Maintain an inventory of re-usable modules. Modules cannot be re-used if developers are not aware of their existence. This inventory should also include re-usable specifications and documentation.

4. Define requirements and functional specifications based on existing re-usable functions. This facilitates the rapid demonstration of the solution and maximizes the possibility for re-using existing functionality. By re-using specs and modules, we will reduce the cost of designing and developing these functions for every application.

SUMMARY

We have discarded proven solutions because newer Object Oriented technologies have claimed to be re-usable but they only provide re-usability at the individual object level (e.g. drop down box). Re-using Object functionality requires a proactive effort on the part of IT organizations but the benefits can be significant. If we implement these recommendations, we will be following the example of the manufacturing industry by assembling components to build systems and reducing dependence on the creative skills of craftsmen.

Finally, we must keep in mind that IT is still a very young profession. Best practices and the associated roles and responsibilities are still evolving in response to rapidly changing technology. Management processes must also be implemented or enhanced to implement this solution.

COMPANY OVERVIEW

CAI is a global IT application management and outsourcing firm. We have been providing application support services to our client base for more than 25 years. Our application support service offering focuses on executing robust processes and management techniques that result in quality applications and, most importantly, reduced IT spend. Through field-proven processes, CAI continuously reduces risk, inefficiency, and costs from the IT application support effort. With more than 2,500 associates located throughout the United States, Canada, Europe, and the Asia Pacific region, CAI is uniquely positioned to provide our customers with 20% or more in hard dollar savings.