



WHITEPAPER

www.claysys.com

CLAYSYS, INC.

BY VINOD THARAKAN
AND SHAJI RAMAKRISHNAN

THE FUTURE OF ENTERPRISE APPLICATIONS

Table of Contents

- 1. Fundamental shifts driving imminent change for Enterprise Software Applications..... 3**
 - 1.1. The emergence of the Cloud Computing paradigm 3
 - 1.2. The key benefits of the shift to the cloud..... 3
 - 1.3. The emergence of ‘No-code’ metadata driven application platforms 4
 - 1.4. The key benefits of ‘No-code and Metadata’ driven Application Development 4
- 2. The current state of Enterprise Applications..... 5**
 - 2.1. Total Cost of Ownership 5
 - 2.2. Change Management 5
 - 2.3. Challenges in ‘doing more with less’ 6
- 3. The Enterprise Applications Opportunity today 6**
 - 3.1. Doing more with less 6
 - 3.2. Adapting to change efficiently 6
 - 3.3. Adopting the best aspects of Cloud computing 7
 - 3.4. Planning for the future 7

1. Fundamental shifts driving imminent change for Enterprise Software Applications

1.1. The emergence of the Cloud Computing paradigm

What is Cloud Computing?

As per the Wikipedia definition, *“Cloud computing is internet-based computing in which large groups of remote servers are networked to allow sharing of data-processing tasks, centralized data storage, and online access to computer services or resources. Clouds can be classified as public, private or hybrid”*.

Practically Cloud computing is simply the ongoing shift where organizations and enterprises that manage their IT Infrastructure and IT Applications within a Datacenter and Network maintained and managed by the organization, are moving to a model where the physical IT Infrastructure is owned, and maintained by the Cloud Service Provider (CSP), and the organization or enterprise rents or leases the required resources on both the Infrastructure side and the applications side, to manage their IT Automation needs.

1.2. The key benefits of the shift to the cloud

- The cost of managing the IT Infrastructure and Applications are reduced significantly due to the economies of scale that allow the CSP to offer very competitive prices for the IT Infrastructure and Applications as compared with the costs of managing it internally.
- The CSP's provide an On-demand resource that allows an organization to scale up and scale down the CSP services rented, to meet the evolving business requirements, thus also transforming IT Expenses to an Operational Expense instead of a Capital Expense.
- While 'Security' is a contentious issue, it can be argued that the CSP's have a more efficient security model, as they can invest into the evolving best practices and apply them across all their clients infrastructure on an ongoing basis, a lot more efficiently than in-house IT keeping up with the evolving security threats. The major security breaches and data losses across large US companies in the last year or so, support this position that a CSP is probably better suited to managing security, than each individual organizations IT Department.
- Speed to market is another key advantage that organizations can leverage through CSP's as it would take minutes to spin up new server capacity, or adopt new business applications. The 'try before you buy' approach for cloud business Enterprise Applications also allow an organization to test the viability of adopting innovative new business applications at practically a negligible cost, as compared to the current model of investing into expensive business applications, and waiting for months or years before getting a chance to review the performance of the applications in meeting business requirements.

1.3. The emergence of 'No-code' metadata driven application platforms

What is 'No-code' application development?

No-code Application Development is an approach to building software business applications at a level of abstraction that makes it easier for 'Power Users' or 'Business Users' to create and maintain their own applications, without needing to depend on specialist 'Software Programmers' for translating the business requirements into working functionality within an application.

Good examples of 'No code' applications are Microsoft Excel and Microsoft PowerPoint, in fact everything within the MS Office suite of applications. Microsoft SharePoint and Force.com from Salesforce are other examples of no-code driven application development. Businesses do not depend on IT support to build whatever they want in Excel or PowerPoint, and can do it on their own. Similarly with adopting SharePoint or Salesforce, IT only has to facilitate provisioning access to these platforms to their Users, after which the Power Users or Business Users can manage their requirements directly through a 'Self Service' model.

The success of such no-code models are proven by simply contemplating the option of switching off access to Excel or other spreadsheet software within an organization as an example. In spite of organizations having other major enterprise applications like SAP or Oracle implemented, they cannot function without spreadsheets, as the operations of an organization could literally come to a standstill if using spreadsheets was not allowed.

No-Code Application Development leverages a 'Metadata' driven architecture for creating the functionality required by their Users. Metadata is simply data that describes data, and a good example is HTML. HTML is just text described using certain text tags that describe the formatting of a web page. A Web Browser is served an HTML file from a Web Server, after which the textual HTML data is rendered into the Web Page that the user interacts with in their browsers.

1.4. The key benefits of 'No-code and Metadata' driven Application Development

- When the creation of an application is possible through a 'Drag and Drop', 'Point and click' type of interface, without needing to write low level cryptic software code, the number of people who can participate in the creation of business applications within an organization increase exponentially.
- Such no-code metadata driven application development models enable you to create the functionality you want exponentially faster, and it is proven to be at least 8 to 10 times more productive as compared with the custom coding model.
- Avoiding the layer of pushing requirements down to the 'Software Programmers' also reduces the changes or mis-interpretation of business requirements of ongoing change management significantly, thus improving the turnaround time of getting working functionality available to the business users.
- Defining the structure and logic of a software business application in 'Metadata' also ensures that the application is 'Futureproof', as it is now possible to have the applications created with earlier

versions of the no-code platform, continue to work with the latest and greatest versions of the no-code platforms, simply because the newer versions are backward compatible with the metadata of the historic versions, thus allowing the old applications to continue working in the latest versions of the no-code environments. Good examples of this are the latest Microsoft Excel versions continuing to support and work with old versions of Excel files created with Office 95, which is almost 20 years old. Another good example is HTML pages created way back in the nineties, continuing to work and render just fine in modern web browsers like Google Chrome, which did not even exist at the time the early HTML pages were created.

- No-code metadata driven applications also have an architecture better suited to ensuring better security, and the application of ongoing security updates in a feasible and practical manner. Since the definitions of all the Forms and Modules in an Enterprise application are maintained as metadata, there is a single Forms or UI Rendering Engine that renders the front end of the Forms at runtime. This architecture allows for all security hardening to be done at the Form or App Rendering engine level, to ensure that every Form rendered inherits all the web form security best practices. As new vulnerabilities are identified, the Form Rendering engine can be updated to defend against the new vulnerabilities, after which every existing form and new form, created with it inherits the same protection. This security model that is available in the no-code metadata driven architecture is not available for the hard coded traditional applications that exist today, so as new vulnerabilities are identified, each and every Form created in .Net or Java has to have its code modified to incorporate the defense's to the newly identified web form vulnerabilities, which in most cases is just not practical or feasible to do.
- Finally, with the structure and logic of a software business application defined in metadata, this architecture enables future renderings of the business applications conceived and built today, on front end platforms that have not even been invented as yet. As such future platforms can simply interpret the 'Metadata', and understand the structure and logic of the underlying business application defined in the metadata, to now support its rendering on any future platforms that have not yet been invented. An example of this concept is books written at a time when electronic devices did not even exist, are now being rendered on eBook devices, simply because the eBook devices support the rendering of the alphabet on an electronic device, thus recreating the experience of reading from an electronic screen, just like you read a book off paper. When the bible was written, nobody envisioned the future existence of eBooks, but today you can read the very same bible on an eBook like the Amazon Kindle.

2. The current state of Enterprise Applications

2.1. Total Cost of Ownership

Organizations and their IT Departments are struggling with the total cost of simply maintaining the existing suite of Enterprise Applications, as the cost of such maintenance increases every year, without significantly improving the applications or adding additional capabilities to these existing Enterprise Applications.

2.2. Change Management

Organizations have to continuously evolve as they react to market changes or competitor actions and strategies, which in turn require them to modify the behavior of their existing Enterprise Applications. But introducing changes to the existing Enterprise Applications are a very painful, time consuming and expensive

process, which hold back most organizations from reacting in an agile manner to the evolutions in the marketplace.

Since the existing paradigm of Enterprise Applications being built on custom coded platforms requires highly skilled programmers, it is also an ongoing challenge to maintain these applications with new programmers, when the experienced programmers who built or customized the applications leave the organization. This problem introduces an additional layer of complexity and risk when attempting to change or modify the existing business applications.

2.3. Challenges in 'doing more with less'

So while IT Departments within organizations struggle to simply maintain the status quo with their existing Enterprise Applications, they are also challenged to react and modify the Enterprise Applications using the same resources and budget constraints that exist. This situation arises not by choice, but due to a need for every organization to evolve based on evolving market trends and competitor actions, and such evolution can only be implemented operationally by applying the required changes to the existing Enterprise Applications, to support the new requirements specified. This ongoing challenge to 'Do more with less' is another imperative that justifies the transition to a No-code, Metadata driven model for Enterprise Applications. All the benefits summarized above for the No-code, Metadata driven model of creating and maintaining Enterprise Applications are significantly better suited to the challenges faced by IT Departments today.

3. The Enterprise Applications Opportunity today

3.1. Doing more with less

- If an organization reviews the benefits of a slow but sure transition to the cloud computing paradigm, without needing to change anything else, significant capital and operational cost benefits can be leveraged through such a transition.
- Adopting No-code, Metadata driven models for application development will significantly increase the efficiency and productivity of the existing resources available to an organization, to deliver significantly more functionality and innovative new applications, without needing to invest in advance of tasting any benefits, as they are forced to do today.

3.2. Adapting to change efficiently

- All organizations have to be able to react to market trends and competitor actions efficiently and in a timely manner, to be able to survive and grow their business in this hypercompetitive marketplace in the world today.
- Such agility can only be achieved through leveraging the 'On-demand' capacity model of the Cloud, along with the No-code, Metadata driven application development model. Even if an organization is willing to absorb the cost of incorporating changes to existing Enterprise Applications, they are still faced with extended and usually unacceptable time periods, for realizing the changes required in their Enterprise Applications built with the current custom coded model.

- When departments and power users are presented with a 'Self Service' type option to introduce changes to their Enterprise Applications, this approach also enables delegating the authority and ability to introducing change, to the frontlines of an organization, thus making it more efficient and cost effective, versus the centralized model that exists today, which is painful with high risk, high cost and a lack of predictability. Very often, by the time IT introduces a change to an Enterprise Application as requested by the business, due to the delays and time constraints in implementing the changes, the business need for such a change itself would have evolved with additional changes required, or would have been made redundant due to the delays in incorporating the required changes.

3.3. Adopting the best aspects of Cloud computing

- Cloud Computing really presents a tremendous opportunity to Enterprise IT Departments, to enable them in being more agile, efficient and cost effective in meeting the business requirements that they support.
- While there is no need to plan a 'big bang' type migration to the cloud, it just makes sense for organizations to plan a transition to the cloud, as the opportunity presents more agility, lower cost, and ongoing efficiencies in meeting the evolving business requirements. E.g. If an application is designed to have a User load of a 1000 users, and is therefore sized to work on a smaller server to suit the required capacity, it would take weeks or months for the organization to increase the server capacity available to the application, if the Users scale to 10's of thousands for reasons that nobody could predict, which is a common occurrence. In the cloud paradigm, IT could spin up additional server capacity in minutes. Also if the peak load of Users is only for specific periods of time, IT can spin up additional capacity only for those days when high workloads are expected, as in the cloud you pay for the capacity you use by the hour.

3.4. Planning for the future

- While it is easy to review and understand the potential benefits of Enterprises leveraging the value of Cloud Computing and the available No-code, Metadata driven application development models, it is not practical to rip out huge existing Enterprise Applications and replace them with new versions built on the latest best practices of cloud computing and No-code, Metadata driven Application Development models.
- There is a lot of time and effort that went into building and implementing existing Enterprise Applications, and though the tools available to create newer, better versions of such applications exist today with the cloud and no-code app dev models, the time and effort that would go into a re-build of such entrenched applications would introduce risk and significant migration costs, that could make this process challenging.
- So the recommended approach would be to first classify existing Enterprise Applications into what is termed 'Systems of Record' and 'Systems of Engagement'. Systems of Record are the entrenched Enterprise Applications like an SAP or an Oracle Financials or a Core Banking system, which over time incorporate the core operational and administrative needs of an organization. Systems of



Engagement on the other hand are the systems developed to extend the access to Systems of Record, to new channels and new user communities for the purpose of efficiency and fostering innovation. Below are three good examples of Systems of Record and their complementary Systems of Engagement which are presented in the table below.

Systems of Record	Systems of Engagement
Core Banking Systems used by Bank Employees across branches	Internet Banking Exposed to Banks Customers for online transactions
Financial Systems used by a closed group of internal employees	Procurement process automation and Order management systems exposed to clients, vendors and partners
HR & Payroll Systems used by HR, Finance and Accounting Users within a company	Expense Reimbursement and Absence Request systems exposed to all employees

- Once such a classification between Systems of Record and Systems of Engagement are made, the Systems of Engagement present the most immediate and viable opportunity to adopt the cloud computing and no-code application development paradigms. The key thing to note also are that 90% of the requirements to incorporate changes to Enterprise Applications are typically for the Systems of Engagement. So planning for the opportunities presented today to be applied to the Systems of Engagement within organizations, present the most practical and best return on investment for any organization wanting to start down this path.

For more information please go to www.claysys.com