Platform As A Service Effort Reduction

Aspen Olmsted, Kaitlyn Fulford
College of Charleston
Department of Computer Science, Charleston, SC 29401
e-mail: olmsteda@cofc.edu, fulfordke@g.cofc.edu

***Abstract*— In this paper, we investigate the problem of development costs in Platform as-a Service (PAAS) cloud-based systems. We develop a set of tools to analyze the size of code executed to support features in the PAAS. In this research, we specifically focus on stable open source platforms to ensure as much of an equivalent offering from each platform. A distinction is made between PAAS and Platform Infrastructure as-a Service (PIAAS). The focus of the paper is on the features provided to the developer that are not provided by traditional network operating systems. Our study demonstrates a cost savings of nearly thirteen million dollars to develop the application services provided by a typical PAAS.**

Keywords-PAAS; cloud computing; CRM

I. INTRODUCTION

In this work, we investigate the problem of estimating the cost of developer services provided by a platform as a service (PAAS) cloud-based system. In traditional client-server architectures, developers spend a great deal of their effort developing functionality that is not specific to the business domain where the application will operate in.

Cloud computing has traditionally been made up of three broad categories of offerings:

- Software-As-A-Service (SAAS) – This category includes applications that run in a Web browser and do not require any local software and hardware besides the Web browser and Internet connection. Examples of software in this category are Google Docs [1] and Microsoft Office 365 [2].
- Infrastructure-As-A-Service (IAAS) – This category includes virtualization software that allows an operating system to be run in the cloud. Typically, the user will pick a hardware configuration and install an operating system into the virtual hardware configuration. IAAS was designed to free the user from the purchase of hardware and allow for hardware upgrades easily. Examples of IAAS offerings are Amazon EC2 [3] and Rackspace [4].
- Platform-As-A-Service (PAAS) – This category includes pre-build components that a developer can use when developing a cloud application. The goal of PAAS is to allow the developer to focus on the development of a solution for the business functions and not software functions that span many application domains. A good example of PAAS is force.com where the developer is provided many of the essential parts of an application out of the box.

Over the years, software development has matured to allow the developer to spend a larger percentage of their development time on the business problem instead of the infrastructure for the application. In the early days of programming, each instruction the programmer wrote matched an instruction in the hardware. The late 1980s and 90s were dominated by 3rd generation languages such as C, PASCAL or ADA where each instruction written by the developer was compiled to many machine instructions. The 21st century has been dominated by byte code compiled languages that have runtime engines that execute the code on different hardware platforms. The Java Runtime Engine (JRE) and the Microsoft .NET Runtime Engine (.NET) are the most dominate examples of the byte code engines that free the developer from thinking about the underlying hardware. PAAS is the next evolution in freeing up the developer times, so they can focus on the problem they are trying to solve instead of the technical plumbing required for the solution.

The organization of the paper is as follows. Section 2 describes the related work and the limitations of current methods. In Section III, we document typical services provided. Section IV analyzes different PAAS providers and the services they provide. Section V explores the alternative costs to develop the individual services. We conclude and discuss future work in Section VI.

II. RELATED WORK

The NIST Definition of Cloud Computing defines PAAS as "the capability provided to the consumer [...] to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages, libraries, services, and tools supported by the provider. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly configuration settings for the application-hosting environment" [1]. In the same document, they define SAAS and IAAS similarly to our definitions in the introduction.

Kolb and Wirtz [2] investigate ways to construct applications for the cloud that are portable across different PAAS providers. Their work assumes lower level services in the offerings than our work. We are less interested in maintenance costs to move platforms as we are in startup costs for Greenfield Engineering. In software engineering,

 ISBN: 978-1-61208-529-6

Greenfield Engineering occurs when you are starting from scratch, or you are re-engineering your product on a different architectural paradigm where you cannot port your current code base.

Baliyan and Kumar [3] explore how services provided by a PAAS provider effect the Software Development Lifecycle (SDLC). Again, in their work they consider just a few services. In our work, we think about many more services. The larger perspective on service would have an even greater impact on their work.

In our model of services, end-users can create new objects, new forms for data entry, new reports to display the data in detail and aggregate form and new dash boards. Ng [4] looks at PAAS as a model for deploying end-user programming through a model of Tasks. The programming model provided by the platforms in our study has demonstrated success in allowing end-users to extend the application.

Boehm, Clark, Horowitz, Westland, Madachy and Selby [5] developed an algorithm to estimate effort for a software engineering project. The algorithm uses variables that represent programmer and programming experience required in the project. For this study, we used the "nominal" value for each variable to get an average cost. Madachy [6] provides an online tool to calculate the effort including maintenance over the life of the software.

III. PAAS SERVICES

With PAAS, the developer does not need to be concerned with the operating system on which the specified platform runs. For example, the platform will provide a service to save a file and the developer does not need to worry what operating system the platform is running on. We group the service offerings into two distinct categories:

A. Infrastructure Services

- Node Configuration – This service allows the end user to modify configuration settings to allow the system to scale to handle larger or smaller workloads by adding or removing nodes, storage or Central Processing Units (CPUs). This service allows the implementation to start with minimal hardware to save costs during start-up. Additional resources can then be added as the application user base grows without the need to re-engineer the application.
- Load Balancing – This service allows the end user to setup multiple systems to ensure uptime when loads are higher, or network partitions occur. Each system is an exact replica and the load will be distributed across the replicas. The application will need to be designed properly for replication. The system must not store resources in a specific replica. Each request could be sent to a different replica. Both persisted data and session state should be stored in the database service.
- Logging – The logging service allows an audit log to be enabled to help diagnosis application and platform issues. The service should allow the log to be toggled on and off so that space is not wasted when an audit is not needed. Ideally, there will be different granularity of audits available, such as errors, warnings, and information.
- Database – The database service allows the application to persist data across executions of the application. Traditionally, this has been a relational database such as Oracle [11] or MySQL [12] but may be a NoSQL [13] database that is better at distribution. The database service should provide Create, Read, Update and Delete (CRUD) services and potentially transaction support.
- Scheduled Jobs – This service allows bulk operations to be scheduled at specific and recurring timeframes. Example jobs include sending out bulk emails, updating de-normalized database fields, and communicating with external systems. Often, this service is delivered through a cronjob interface where jobs can be schedule down to the specific second of each hour.

B. Application Services

- Authentication – The authentication service provides a way to define users and allow authentication in the application being developed. Ideally, this would provide both the administrative tool for creating users and groups along with the user interface the end users interact with to authenticate themselves. The authentication should provide multifactor authentication where something the end user knows along with someplace they are or something they have.
- Authorization – The authorization service provides a way to define which users can see different data, forms, and reports in the application. The authorization service should provide an administrative tool to assign access permission to both users and groups to objects created in the system. The objects should be both standard objects and custom objects defined by the developer and end users.
- Rule Engine – A rule engine allows for customization of correctness rules at implementation time. Business rules control organization policies that may change often and should not be coded in the software solution.

 ISBN: 978-1-61208-529-6

- Workflow – This service provides for several discrete application steps to be sequenced together. Often, a human interaction (Approval) is part of the workflow.
- Bulk Email – Bulk email allows for email marketing with proper adherence to Email SPAM rules [14]. Bulk email may be used for attracting or recruiting new customers or confirming transactions with current customers.
- Importing – An importing feature allows the end user to import new instances of objects into the platform. Ideally, this would allow data from several different data formats including Comma Separated Values (CSV) and Microsoft Excel format. The tool should provide a validation step so that imported data does not corrupt the current database.
- Exporting – An exporting feature allows the end user to dump instances of the objects into an external file such as a Comma Separated Values (CSV) or Microsoft Excel formatted file. The tool should allow a Query by Example (QBE) where novice users can visually build export queries and see the results in the application.
- Activity tracking – Activity tracking allows for linking of phone calls, emails, meetings, and notes to objects persisted by the application. Activities may be originated in an external system with an interface to the new system that is being built. An example could be a Web browser extension that allows emails in a Web email application to be linked to a related activity to an object in the new system.
- Object Customization – Object custom allows end users to add additional data to be collected in the application without changing the source code. Most enterprise systems require some form of customization either through integration to external systems or enhancements to specific features in the current system. Object customization allows the end user to make the changes without needing the software to be modified at each individual enterprise.
- New Object Creation – New object creation services allow end users to define new objects to store data that is collected in the application. Like with object customization, new object creation can be used to customize the software without changing the source code. Often, the new objects need to relate to a current object in the system. These related objects should seamlessly be displayed in the user interface.
- Detail View – The detail view renders the object details based on the configured layout. The detail view also renders one-to-many related data. The related data is often rendered in tabs.
- Edit View – The edit view renders an editable object screen based on the configured layout. The edit view is used to modify one specific object and potentially related objects.
- Data Update – The data update service provides a CRUD interface to a backend data store. The data update service abstracts the vendor specifics of the back-end data store services and allows business rule hooks to fire on the CRUD operations.

TABLE 1. APPLICATION SERVICES BY PLATFORM

Service	Salesforce	Zoho	SugarCRM	SuiteCRM	vTiger	Heroku
Authentication	✓	✓	✓	✓	✓	✓
Authorization	✓	✓	✓	✓	✓	
Rule Engine	✓		✓			
Workflow	✓	✓	✓	✓	✓	
Bulk Email	✓	✓	✓	✓	✓	
Activity tracking	✓	✓	✓	✓	✓	
Object Audit	✓		✓	✓	✓	
Importing	✓	✓	✓	✓	✓	
Exporting	✓	✓	✓	✓	✓	
Object Customization	✓	✓	✓	✓	✓	
New Object Creation	✓	✓	✓	✓	✓	
User Interface Customization	✓	✓	✓	✓	✓	
Multi-Select Fields	✓	✓	✓	✓	✓	
Report Display	✓	✓	✓	✓	✓	
Report Creation	✓	✓	✓	✓	✓	
Dashboard Display	✓	✓	✓	✓	✓	
Dashboard Creation	✓	✓	✓	✓	✓	
Web-services	✓	✓	✓	✓	✓	
Mobile Application	✓	✓	✓		✓	
Partner Portal	✓					
Customer Portal	✓		✓	✓	✓	
Anonymous Sites	✓					✓
Price per user/month	$25	$35	$65	N/A	N/A	N/A

 ISBN: 978-1-61208-529-6

- User Interface Customization – The user interface customization service allows for forms in the application to be modified by the end users without changing the source code. This is often required to allow implementations to vary slightly by collecting custom data.
- Multi-Select Fields – Multi-select fields are a way to simplify end user customizations. A multi-select field represents an easy way to store a one-to-many relationship of data without the need of adding new objects. Multi-select fields also save on the number of tuples stored in the system. Often, cloud providers charge for data storage based on the number of tuples. [7]
- Report Display – The report display service allows execution of pre-defined reporting queries. The report display should prompt the user with replaceable run-time parameters. The report should be exportable to pdf and spreadsheet formats. Ideally, there would be a scheduling service where the report parameters would be based on the run date. For example, a start date parameter should be replaced based on an offset from the date the report is run.
- Report Creation – The report writer service allows both the developer and the end users to define management information system (MIS) reports that can be run and customized by the changing of run-time parameters. Typically, this includes both tabular reports that group rows of records with aggregate calculations and cross-tab reports that aggregate values based on the intersection of the row and column.
- Dashboard Display – The dashboard display service renders dashboard charts and allows them to be refreshed automatically. The dashboard is a graphical display of a metric the organization wants to measure.
- Dashboard Creation – Dashboards allow both the developer and the end users to define graphical dashboards that allow visualization of data stored in the application. Dashboards typically are bar or pie charts and are updated several times an hour.
- Mobile Application – A mobile application allows end users to perform create, read, update, and delete (CRUD) operations on objects stored in the application without the need of creating custom mobile applications. Similar to the detail and edit view services above, any object in the system should be visible and editable.
- Partner Portal – A partner portal is a service to provide pages, forms, reports and dashboards to authenticated users with a lower training level. Typically, these are users that use the application infrequently compared to an employee.
- Customer Portal – A customer portal is a service to provide custom pages and forms to authenticated users with no training required. The service is intended for customer self-service sites where the customer can identify themselves and perform transactions.
- Anonymous Sites – The anonymous site service allows development of pages and forms to unauthenticated users. This is typically the part of an organizations website where customers do not need to identify themselves.

IV. PLATFORM ANALYSIS

In this study, we analyze several PAAS providers including Salesforce [8], Zoho CRM [9], SugarCRM [10], SuiteCRM [11], vTiger [12] and Heroku [13]. We chose the first five platforms because they each provide many of the services we discussed in detail. The last platform was added to show the difference between PAAS and PIAAS offerings. Each of the first five PAAS offerings was developed as Customer Relationship Management (CRM) system. The CRM vertical market software space requires integration with Enterprise Resource Planning (ERP) systems. The integration requirement led the CRM vendors to develop their products as platforms instead of just the vertical market products. TABLE 2 shows the distributed services offered by each platform. The Load Balancing service is marked for the three PHP [18] platforms because the state of the session is stored in the database. Having the session state stored in the database allows additional business tier servers to be added to the configuration in the cloud. Though only Heroku has a graphical user interface (GUI) to manage node configuration, the PHP solutions can be hosted by an IAAS provider that provides the feature. TABLE 1 shows the

TABLE 2. DISTRIBUTED SERVICES BY PLATFORM

Service	Salesforce	Zoho	SugarCRM	SuiteCRM	vTiger	Heroku
Node Configuration						✓
Load Balancing			✓	✓	✓	
Logging	✓		✓	✓	✓	✓
Database	✓	✓	✓	✓	✓	✓
Scheduled Jobs	✓	✓	✓	✓	✓	

 ISBN: 978-1-61208-529-6

application services offered by each platform. The final row shows a cost per user if the PAAS provider is providing both the infrastructure and the application services.

V. EFFORT STUDY

To calculate the effort savings provided by the different PAAS service providers, we calculated the source lines of code (SLOC) in a stable platform release. For this study, we choose to use the SuiteCRM [10] systems as our model. SuiteCRM is open source software, so we had access to the source code developed to provide the platform.

SuiteCRM is written in the PHP programming language using a MySQL database as its persistence layer. Using the debugger extension xDebug [13], we are able to trace all lines of code executed on the server when interacting with the application. xDebug creates a trace file with these lines of code. We developed tooling to parse the trace file and store the data in a MySQL database based on the function executed.

Because of the nature of Web application architectures, a single round trip from the Web browser to the Web server will often execute two distinct sets of functionality. For example, when a user enters their login credentials, the post to the server authenticates the user and then executes the code to display the homepage of dashboards. Our tooling allows a trace to add to the functional cost or subtracted from the functional cost. In the earlier example, we trace the combined functionality and then subtract the individual functionality of building the home screen.

For the study, we wanted the cost for local application software engineers in the Charleston, SC area. The Bureau of Labor Statics [14] estimated the average cost for an application software engineer is $96,200/year. Hadzima [15] estimates the cost of an employee's benefits and taxes at between 25% and 40% of base salary. On top of the salary cost, the employer must pay for rent for office space, equipment, recruitment, training, etc. For our study, we are estimating the hourly cast at $71.50 per hour for an application programmer's time. In TABLE 3 we show the estimated cost to pay an application programmer in the Charleston, SC area to redevelop the functionality provided by the service. We analyzed SuiteCRM and looked at the organized source lines of code (SLOC). SuiteCRM stores the service source code in module folders on the file systems. We counted the executable lines of code and compared to the executed lines of code from the trace. Each trace represented a slightly higher number of lines of code because of shared libraries. We felt it was not appropriate to count all the shared lines of code per service, but we also felt it was not appropriate to ignore them completely. We decided to take the average between the two-line counts. We plugged the average number into the Constructive Cost Model (COCOMO) II formula [26] with our local application programmer cost of $71.50 per hour. The fifth column in TABLE 3 shows the cost per service and the total cost of all services. We eliminated a few services from the study as the source code was not available. The table does not show the cost of the infrastructure services. The infrastructure services can be provided by an IAAS provider if the development is done to leverage the services.

VI. CONCLUSION

In this paper, we analyzed the programming effort required to reproduce services provided by a cloud PAAS provider. Our solution utilizes two methods to estimate the number of lines of code requried for a service; SLOC and an execution trace. We utilize an average of the two methods to apply the COCOMO II costing algorithm. Our study demonstrates a cost savings of nearly thirteen million dollars. The savings comes from not needing to develop the application services provied by the PAAS providers in our study.

In this research, we focused on application services provided by a PAAS. Future work needs to study the infrastructure services costs and the application development knowledge required to leverage the provided distribution services.

TABLE 3. COST PER SERVICES

Service	SLOC	Trace	Average	CoCoMoII
Authentication/ Authorization	1156	1437	1297	$ 590,131
Workflow	954	1146	1050	$ 467,789
Bulk Email	702	1054	878	$ 384,246
Activity tracking	1178	1302	1240	$ 561,674
Object Audit	656	873	765	$ 330,225
Importing	1654	1857	1756	$ 823,478
Exporting	945	1246	1096	$ 490,375
Object Customization	2164	2874	2519	$ 1,224,557
New Object Design	1474	1826	1650	$ 768,981
Detail View	291	464	378	$ 152,095
Edit View	1828	464	1146	$ 515,031
Data Updates	656	989	823	$ 357,860
User Interface Customization	2073	2482	2278	$ 1,096,353
Multi-Select Fields	402	512	457	$ 187,395
Report Display	1912	2356	2134	$ 1,020,384
Report Creation	2957	3345	3151	$ 1,566,362
Dashboard Display	1342	1672	1507	$ 696,016
Dashboard Creation	1874	2198	2036	$ 968,972
Web-services	986	1822	1404	$ 643,884
Total				$ 12,845,808

 ISBN: 978-1-61208-529-6

REFERENCES

[1] Google, "About Google Docs," 2017. [Online]. Available: https://www.google.com/docs/about/. [Accessed 10 02 2017].

[2] Microsoft, "Office products," 2017. [Online]. Available: https://products.office.com/en-us/products. [Accessed 10 02 2017].

[3] Amazon Web Services, Inc, "Amazon Elastic Compute Cloud - Virtual Server Hosting," 2017. [Online]. Available: https://aws.amazon.com/ec2/. [Accessed 10 02 2017].

[4] Rackspace, "Rackspace.com - Rackspace® Managed Cloud," 2017. [Online]. Available: https://www.rackspace.com/. [Accessed 10 02 2017].

[5] P. Mell and P. Grance, "The NIST Definition of Cloud," 09 2011. [Online]. Available: http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf. [Accessed 07 09 2016].

[6] S. Kolb and G. Wirtz, "Portability in Platform as a Service," in *2014 IEEE 8th International Symposium on Service Oriented System Engineering*, Oxford, United Kingdom, 2014.

[7] N. Baliyan and S. Kumar, "Towards Software Engineering Paradigm for," in *2014 Seventh International Conference on Contemporary Computing*, Noida, India, 2014.

[8] J. Ng, "Extending the Cloud From an App Development Platform into a Tasking Platform," in *2015 IEEE World Congress on Services*, New York, NY, 2015.

[9] B. Boehm, B. Clark, E. Horowitz, C. Westland, R. Madachy and R. Selby, "Cost models for future software life cycle processes: COCOMO 2.0," *Annals of Software Engineering*, vol. 1, no. 1, p. 57–94, 1995.

[10] M. Ray, "COCOMO II - Constructive Cost Model," 2016. [Online]. Available: http://csse.usc.edu/tools/COCOMOII.php. [Accessed 07 09 2016].

[11] Oracle, "Oracle Database," 2017. [Online]. Available: https://www.oracle.com/database/index.html. [Accessed 10 02 2017].

[12] Oracle, "MySQL Database," 2017. [Online]. Available: https://www.mysql.com/. [Accessed 10 02 2017].

[13] Wikimedia Foundation, Inc, "NoSQL," 2017. [Online]. Available: https://en.wikipedia.org/wiki/NoSQL. [Accessed 10 02 2017].

[14] Wikimedia Foundation, Inc, "Email spam," 2017. [Online]. Available: https://en.wikipedia.org/wiki/Email_spam. [Accessed 10 02 2017].

[15] A. Olmsted and G. Santhanakrishnan, "Cloud Data Denormalization of Anonymous Transactions," in *Cloud Computing*, Rome, Italy, 2016.

[16] Salesforce.com, inc, "Run your business better with Force.," 2006. [Online]. Available: http://www.salesforce.com/platform/products/force/?d=70130000000f27V&internal=true. [Accessed 03 02 2016].

[17] Zoho Corporation Pvt. Ltd, "Zoho CRM is ready," 2016. [Online]. Available: https://www.zoho.com/crm. [Accessed 07 09 2016].

[18] SugarCRM, "Discover a different kind of CRM," 2016. [Online]. Available: http://www.sugarcrm.com/. [Accessed 07 09 2016].

[19] SalesAgility, "SuiteCRM – CRM for the world," 2016. [Online]. Available: https://suitecrm.com/. [Accessed 07 09 2016].

[20] vTiger, "Grow sales, improve marketing ROI, and deliver great customer service," 2016. [Online]. Available: https://www.vtiger.com/. [Accessed 07 09 2016].

[21] Salesforce, "Cloud Application Platform," 2016. [Online]. Available: https://www.heroku.com/. [Accessed 07 09 2016].

[22] The PHP Group, "About PHP," 2017. [Online]. Available: http://php.net/. [Accessed 10 02 2017].

[23] D. Rethans, "Xdebug - Debugger and Profiler Tool for PHP," 2016. [Online]. Available: www.xdebug.org. [Accessed 07 09 2016].

[24] Bureau of Labor Statisics, "Occupational Employment Statistics," 2016. [Online]. Available: http://www.bls.gov/oes/current/oes_16700.htm. [Accessed 07 09 2016].

[25] J. Hadzima, "How Much Does An Employee Cost?," [Online]. Available: http://web.mit.edu/e-club/hadzima/how-much-does-an-employee-cost.html. [Accessed 07 09 2016].

[26] R. Madachy, "COCOMO II - Constructive Cost Model," [Online]. Available: http://csse.usc.edu/tools/COCOMOII.php. [Accessed 10 02 2017].

 ISBN: 978-1-61208-529-6