International Journal of Engineering & Technology, 7 (2.32) (2018) 86-90

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Worked out Results of IPCCR Framework for AMS Projects

Srinivasa Rao Kosiganti[1], Dr. Y. Prasanth[2]

[1]Research Scholar, KL University, Vijayawada
[2]Professor, Research Supervisor, KLU, Vijayawada, Andhra Pradesh, India.
**Corresponding author E-mail: srinikosi@gmail.com*

Abstract

The IPCCR framework that was designed to reduce CAPEX and OPEX of Application Support and Maintenance Projects, has helped to understand Incident, Problem, Change, Configuration and Release and directly impacts the costs that are accrued. Using Proper Incident Management, Problem Management, Change Management, Configuration Management and Release Management, which are the key ingredients of ITIL V3.2 and DevOps reduces the costs of Capital Expenses and Operational Expenses. The important concept of Known Error Database will subsequently reduce the Operational Expenses as much as possible.

Keywords:*IPCCR, AMS, CAPEX, OPEX.*

1. Impact of IPCCR Framework

In this paper, I tried to introduce Incident Priorities, Severities and when Severities are high, it leads to Problems and to tackle that, Problem Management been used. Once a Change Requirement mode is understood, a possibility of CAB(Change Advisory Board) can be set up, which not only strengthens the quality process but also increases revenue to the Vendor. Proper Configuration Management on one server reduces a lot of redundant servers where similar kind of change requests worked out from different servers, be tackled easily using one server. Configuration Management allows to reduce that. Having Proper Release Management in terms of Patches, Parallel Releases, would reduce the burden of operational expenses to the optimum level because parallel releases can be configured on to the same servers and the same personnel can work as the artifacts are same, and design and coding of the enhancements or incidents would take less effort.

2. Definition of Incident Priorities/Severities (P1, P2, P3 and P4)

This section details incident Priorities/Severities and related information.

3. Severity Levels under Steady State

The following table provides a sample definition of Priority/Severity levels associated with an incident or a problem. The tickets based on the Priority/Severity are executed through the L1/L2/L3 support process and are resolved based on the timeline agreed for each severity.

Table 1: Severity Levels

Priority/Severity Level	Priority/Severity Classification	Problem Characteristics
1 (Critical)	Severe business disruption	Service is not available. Major impact on users and business. 1. The problem will result in a major loss of revenue or a major cost to Client 2. There is the potential of a health, safety or security issue to occur, or it has already occurred. 3. A system, application or function is completely unavailable, severely corrupted or severely degraded for a large number of authorised users.
2 (Urgent)	Major business disruption	Incidents that cause significant impact on users and business and no workaround are available. 1. A system, application or function is completely unavailable, severely corrupted or severely degraded for a limited number of authorised users. 2. A critical transaction or batch application failure has occurred. 3. An entire non-critical department or building is unable to function due to service unavailability.
3 (Normal)	Business disruption	1. Default severity. Incidents that cause significant impact and a workaround are available. 2. A moderate impact problem affecting a small non-critical group of authorised users. 3. A short-term workaround has been made available to provide an acceptable level of service to authorised users.
4 (Low)	Minor disruption	Incidents that cause no immediate impact to service availability or performance. There is a failure in the service but can easily be bypassed. 1. A minimal service impact to one or more users or their performance is impaired, but there is no Client business impact.

4. Effort Estimation for Application Maintenance Projects and Pricing Consideration for AS-IS and TO-BE Scenarios

1. **AS- IS(To have Comparison Results):** For the Sample Application Australia Communications, the FTEs considered earlier were: 11 for 10 applications; (This is a suite of applications taken for experimentation which has followed the routine estimation methodology which are been published in paper listed in the References * Staffing Software Maintenance and Support Projects by **Jai Asundi, Sumit Sarkar, University of**

Texas, Dallas (C) 2005 IEEE for Staffing). This is used for Comparing Results with To-Be Framework.

Table 2: Base FTEs for As-Is

Application Name	Criticality	Base FTE for Year 1* in AS-IS
App 1	Medium	1
App 2	Low	0.5
App 3	Medium	1
App 4	Medium	1
App 5	Low	0.5
App 6	Low	0.5
App 7	Low	0.5
App 8	High	2
App 9	High	2
App 10	High	2

Analogy for Comparing Results: Taken $20 per hour as estimated cost and 252 working days an year and 8 working hours each day.

For As-Is(Applications which were estimated using routine estimation methods described in the Referenced Papers), 11 FTEs were been used, which consumed 11*20*8*252= $443520 - (i)

The no of Servers used in this case were : 5(HP Blade Chassis Servers)

Each Server costs $ 2000.
Total Cost of the Servers were: 2000*5=$10000 (ii)

Now as part of the innovation, Vendor proposed a well-established methodology for estimation of the total effort required to perform Application Support and Maintenance activities for an Application/System or a Group of Applications/Systems (Application Portfolio), known as Due Diligence Cube (DDC) method.

This method uses the following three key drivers to analyse the application behaviour with in an application portfolio:

- Complexity
- Criticality
- Stability

Vendor used this methodology to rank applications from D1 to D9. While D1 is the 'Least complex, least critical and most stable' application, D9 represents 'Most complex, most critical and least stable' application.

The following figure illustrates how the applications are fit in the grid of D1 to D9:

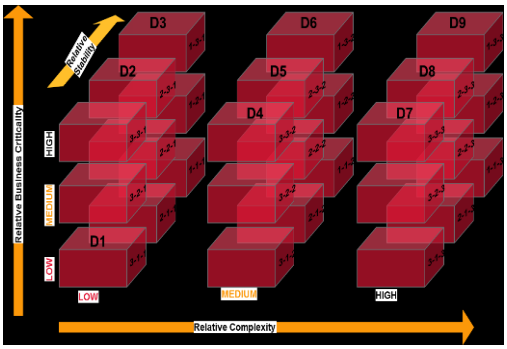

Figure 1: Grid to denote Complexity

The above mentioned three factors were derived from the data provided by the Client, and a Scoring Mechanism is arrived at, for overall effort at an application level. Some of the items which helped Vendor in arriving at the above factors are either given or inferred through other data elements provided by the Client.

The following parameters are considered to arrive at stability of the application:

1. Number of production fixes
2. Number of Severity 1 and 2 bugs worked on
3. Number of System Outages during last one year
4. Recurring issues with the application
5. Known Error Database used for Repetitive bugs
6. ITIL V3.2 framework been used for Service Operations
7. Reliability issues with the application
8. Performance issues with the application
9. Support issues with the application
10. Number of scope changes/CRs/major releases, which have been deployed in last one year. (more the SC/CR/MRs, less the stability of application)

The following parameters were considered to arrive at complexity of the application:

1. Technology Stack - Where Configuration Management, Change Management, Incident Management used extensively
2. Number of external and internal interfaces
3. Application size (number of modules, number of KLOC)
4. Number of source code components
5. Percentage of processing logic
6. Number of batch background job flows
7. Database size
8. Number of Data Stores
9. Business complexity
10. Number of Users and number of Concurrent User Base to be supported
11. Availability of Documentation (for example, execution flows, job flows, System Architecture diagram, Data model, Operational flows, User manual, Data dictionary and so on)

The business criticality of the application is normally provided by the Client or derived from few key parameters like:

1. Support hours coverage (for example, if the application requires 24x5x252 coverage, the likelihood is that it is a business critical application)
2. Application scope-global or local
3. Number of business units it supports
4. Number of languages it supports
5. Business process that the application belongs to (for example, Order Management, Supply Chain Management and so on)

Based on the scores assigned for Criticality, Complexity and Stability for each of the applications, the output of due diligence cube is attained. A representative output of such analysis is shown in the following picture:

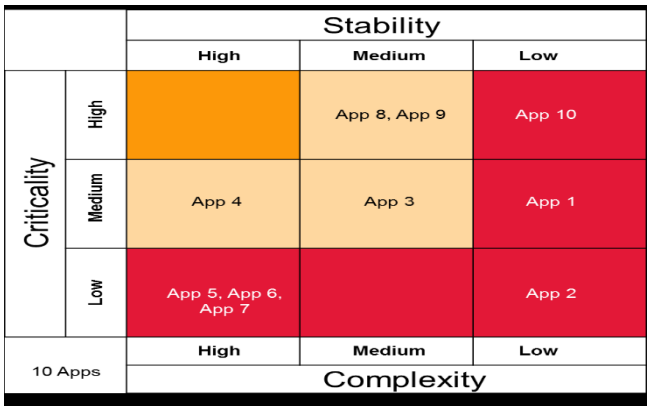

Figure 2: Representative Output of Analysis

After fitting the application in the above grid, the applications are further re-grouped in manageable size (example, the nine groups shown in the above table are re-grouped to G1, G2, G3, if required). The base support required to support an application is calculated depending on the weightages assigned to Criticality, Complexity and Stability and the associated scoring pattern of these parameters. The following table lists details related to arriving at the base effort and complexity of usage of servers and the optimization factor.

5. Illustrations

The following is the weightage table:

Table 3: Weightage Table

Application Characteristic	Weightage Factor
Criticality	40
Complexity	30
Stability	30
Total Score	**100**

The following tables list Criticality, Complexity and Stability scorings:

Table 4: Criticality Table

	High/Medium/Low	Score
Criticality	High	5
	Medium	3
	Low	2

	High/Medium/Low	Score
Complexity	High	5
	Medium	3
	Low	2

Table 5: Complexity Table

	High/Medium/Low	Score
Stability	High	2
	Medium	3
	Low	5

Table 6: Score Table

Based on the portfolio, a table for number of FTE required for a scoring range is attained. A typical scoring range vs. number of FTE is listed in the following table.*

Table 7: Scoring range of FTEs

Score Range	Number of FTE required
= 200	0.25
201-350	0.5
>350	1

*Please note that one need to calibrate this table based on the portfolio for a Client.

Output of DD analysis and the associated effort estimation is listed in the following table:

Table 8: Output of DD Analysis

Application Name	Criticality	Weightage	Complexity	Weightage	Stability	Weightage	Total Score*	Base FTE for Year 1*
App 1	Medium	3	Low	2	Low	5	330	0.5
App 2	Low	2	Low	2	Low	5	290	0.5
App 3	Medium	3	Medium	3	Medium	3	300	0.5
App 4	Medium	3	High	5	High	2	330	0.5
App 5	Low	2	High	5	High	2	290	0.5
App 6	Low	2	High	5	High	2	290	0.5
App 7	Low	2	High	5	High	2	290	0.5
App 8	High	5	Medium	3	Medium	3	380	1
App 9	High	5	Medium	3	Medium	3	380	1
App 10	High	5	Low	2	Low	5	410	1

***Base FTE for year 1** = (Weightage for Criticality x Weightage Factor for criticality) + (Weightage for Complexity x Weightage Factor for Complexity) + (Weightage for Stability x Weightage Factor for Stability)

The Base Effort is the effort required to provide 'Lights on support' to the Application Portfolio. Based on the additional scope involved in the engagement, (for example, minor enhancements, bi-lingual support requirements and so on), the base effort is further calibrated/increased to arrive at the final effort for maintaining the application portfolio.

6. Analogy of Comparative Results:

***** Now using the Framework, Due Diligence Analysis and Grid Method the total FTE count has come down to 6.5.**

The total cost is 6.5*20*8*252=$**262080** **--- (a)**

Between (i) and (a) equations,

There is a drastic reduction of: $**181440**(which is equal to 69%). In terms of Opex, could able to improve upon **69%.**

Now the no of Servers used are: **2**(because Configuration Management, Virtualization been used effectively) in the To-Be Applications apart from the Framework denoted above: (Based on the Stability of the applications - Where High and Medium Stabilized applications deployed on various servers can be Virtualized)

Cost of the Servers in the innovated Framework is: 2*2000=$**4000 (b)**

Between (ii) and (b) equations,

Difference: $10000-$4000=**$6000**
There is an improvement of 66% in terms of Capex.

The following scenarios explain how Vendor validates the base effort arrived at using the DD cube method.

Scenario 1: In case the Client provides the current AM Support Efforts (FTEs) from its existing suppliers and also follows the Configuration Management, ITIL V3.2 framework, Change Management, Incident Management and Virtualization;

- In this case Vendor will follow the approach mentioned above, arrive at its efforts and cross check the efforts/FTEs provided by the Client.
- Vendor will look at the possibilities to reduce the efforts/FTEs with respect to Client provided, as the Client expects reduction of the existing spend
- It also brings reduction in Capex in terms of the no of Servers

Scenario 2: In case if the application portfolio has less number of applications (10-15) and a maximum of one or two technologies (for example, SAP, Oracle) Vendor can follow the approach mentioned in the following list:

- Consider Severity 1, 2, 3 tickets, their response and resolution times.
- Consider Vendor effort guidelines for the given technology.
- Come up with the efforts based on the ticket volume and the efforts norms.
- Come up with how many no of servers been used and can they be striated for virtualization or not;

This approach will help in getting efforts/FTEs and the cost of Servers, required for the Y1 for the application portfolio in scope.

To arrive at the efforts/number of FTEs from Y2 onwards, Year-on-Year (YoY) productivity gains that Vendor can pass back to the Client are calculated and factored in to the over-all effort. Other levers, which would help Vendor in reducing the price of operations for the portfolio, include YoY improvement in the offshore leverage and adjustment to the senior/junior ration of resources working on the project.

Productivity improvement can be achieved in one or more of the following ways:

- The number of tickets will be reduced as the application's stability increases YoY.
- Team members application knowledge will be enhanced as the time progress so that the time required to fix the issues will be reduced, resulting in more number of tickets being fixed with less number of resources.

YoY productivity of the FTEs for a given application will range between 5-10 percent depending on the application's current maturity level (for example, level of automation done, availability of documentation and so on).

7. Establishing Service Level Framework

Service Level Framework helps Vendor to establish a Service Level Management (SLM) team. The framework enables SLM Team to effectively identify the service levels for the Client, by bringing in right balance in implementation of service and associated cost of implementation. It also ensures that both Client and Vendor understand their roles and responsibilities in ensuring service levels are met. Capacity Management is the core of Service Level Framework.

8. Need for Service Level Framework and Service Level Management (SLM) team

The framework should enable a better understanding between Client and Vendor in:

1. Setting up clear service quality expectations and effectively measuring, monitoring and reporting service quality
2. Distinguishing roles and responsibilities clearly for all the stakeholders involved
3. Enabling flexibility for Client and Vendor to reach desired business needs and market conditions
4. Facilitating correct infrastructure sizing, based on defined service levels
5. Mitigating costs of excess or insufficient capacity
6. Maintaining discipline in supporting other parties providing IT services to the Client

I. Activities of Service Level Framework Management team

The following are the key activities:

1. Identify business requirements by working closely with the Client.
2. Translate business requirements in to IT requirements.
3. Gap analysis between business requirements and available services.
4. Establish service specifics such as the scope of services, timelines, operational hours, recovery aspects, service performance and so on.
5. Determine costs of service aligning with service goals and Client expectations.
6. Draft, discuss, refine and agree on SLAs with all the stakeholders of Client to ensure that their requirements are met.
7. Measure and Report SLA performance.

II. Establishing Service Level Framework and Service Level Management Team

1. It involves three following major steps:
2. Assessment
3. Setting up foundation

4. Establishing Framework

Vendor will ensure that appropriate stakeholders' approval is taken before starting the third step. Vendor will also work with the Client in socializing the processes across the organisation of Client.

III.Post Framework Implementation

Vendor will initiate the SLM work post receiving approval from the Client. By initiating, SLM team will start to take alerts when processes are in danger or not up to the expected level due to unexpected bottlenecks in the system. Vendor's SLM team will monitor the demand and levels of SLA, to take appropriate corrective actions, if required.

SLM team will schedule quarterly and monthly review calls with appropriate stakeholders (as per the framework) for performance review of services. SLM will discuss and agree with appropriate stakeholder on the action items based on the feedback from review meetings. Actions will be initiated at appropriate time. SLM will also discuss any unforeseen business demands, which may occur or change in business priorities, which may impact current services and service levels. Based on the agreement with the concerned stakeholder, appropriate action will be initiated.

9. Conclusion

Based on the agreement with Client, Vendor will conduct an audit, based on agreed timeline (6-12 months) to review the Service Level Framework, SLM team functioning and overall adherence to the SLAs. As part of this audit, Vendor will also review the adherence to industry standards and best practices.

References

[1] Selecting optimal maintenance plans based on Cost/Reliability Tradeoffs for Software Subject to Structural and behavioral changes by Vittorio Cortelessa and Raffeala Mirandola, Pasqualina Potena; 1534-5351/10 $26.00 © 2010 IEEE; DOI 10.1109/CSMR.2010.15.

[2] A Framework for Software Maintenance and Support Phase by Zafar Nasir and Abu Zafar, Department of Computer Science, National Institute of Computer and Emerging Sciences, Karachi, Pakistan; 978-1-4244-8003-6/10/$26.00 ©2010 IEEE.

[3] Software Cost Reduction in Practice, by James A. Hage; 0098-5589/89/1200-1638$01.00 © 1989 IEEE.

[4] Delphi Study of Software Maintenance Problems by Sasa Dekleva, DePaul University, Chicago; 0-8186-2980-0/92 $03.00 © 1992 IEEE,

[5] Empirircal Analysis of Team & Application Size on Software Maintenance & Support Activities by Janaid Aziz, Faheem Ahmed, UAE University, Al Ain, United Arab Emirates; Mohammed Shakeel Laghari, College of Engineering, UAE University, Al Ain, UAE; 978-0-7695-3595-1/09 $25.00 © 2009 IEEE;

[6] Software Maintenance: An analysis of Industrial Needs and Constraints by Hazizi, J.F. Voidrot, Matra Marconi Space France; E. Minor, L. Pofelski, Cap Gemini Innovation; S. Blazy, Electricite de France; 0-8186-2980-0/92 SCn.00 0 © 1992 IEEE.

[7] Measuring and Monitoring Software Maintenance Services: An Industrial Experience by Kaan Kurtel, Department of Software Engineering, Izmir University of Economics, Izmir, Turkey; 978-0-7695-5078-7/13 $26.00 © 2013 IEEE; DOI 10.1109/IWSM-Mensura.2013.43.

[8] Software Maintenance Implications on Cost and Schedule by Bob Hunt, Bryn Turner, Karen McRitchie, Galorath Incorporated, California; 1-4244-1488-1/08/$25.00 ©2008 IEEE, IEEE; AC paper#1098, Version 4, Updated 2007:10:26.

[9] A Process Model for the Maintenance of Large Space Systems Software, by Del-Raj Harjani, Jean-Pierre Queille, Matra Marconi Space, France; 0-81862980-0/92 $03.00 © 1992 IEEE.

[10] Software Maintenance Process Model and Contrastive Analysis by Ren Yongchang, College of Information Science and Technology; Liu Zhongjing, Department of Computer Science, China; Xing Tao, Beijing Research Center of Urban Systems Engineering; Chen Xiaoji, Department of Computer Science, China; 978-0-7695-4523-3/11 $26.00 © 2011 IEEE; DOI 10.1109/ICIII.2011.324.

[11] A survey on the Software Maintenance Process by Maria Joao Castro Sousa, Maria Jo20 Castro Sousa, Helena Mendes Moreira, Departamento de Informatica, Portugal; 0-8186-8779-7/98 $10.00 @ 1998 IEEE.

[12] How to save on Software Maintenance Costs, Omnext White Paper, 2010.

[13] Staffing Software Maintenance and Support Projects by Jai Asundi, Sumit Sarkar, Information Systems and Operations Management, School of Management, University of Texas, Dallas; 0-7695-2268-8/05/$20.00 (C) 2005 IEEE.

[14] A task-based approach to improving the software maintenance process by Larry Cousin, Code 3 / HSI Systems, Murray; James S. Collofello, Computer Science Department, Tempe; 0-8186-2980-0/92 SU3.00 Q 1992 IEEE.

[15] Analysis of Maintenance Work Categories Through Measurement, by Allain Abran, Hong, Montreal Trust, Montreal; CH3047-8/9110000/0104/$01.00 (C) 1991 IEEE