



# A Service Oriented Handover Process for Software Development Projects

Asad Masood Qazi

BI Consultant at Hive Worx, Islamabad, Pakistan

Email: asadmasood1@hotmail.com

Mamoona Bashir

Assistant Manager (IT) at PTCL, Islamabad, Pakistan

Email: mamoona.bashir@ptcl.net.pk

**Abstract:** Handover process is considered as a critical process in software development process because of involvement of such stakeholders, which was not part of the process before. Requirements are shifting between the stakeholders. In case of any issue regarding communication gaps, understanding of a problem or any other misunderstanding, maintenance of software could be affected. Integration of service oriented architecture can create the handover distributed, analytical and helpful to all stakeholders. In this paper, we discussed the service oriented handover process in detail and results show that it creates a positive impact on software development projects.

**Keyword:** Handover Process; Service-Oriented Architecture; Service-Oriented Software Engineering; Decision Making

## 1. INTRODUCTION

Software development and software engineering processes are now a days moving towards service-oriented domains. From requirements transferring to tasks transferring and from operational activities to the strategic level of organizations, they need a proper service-oriented approach of software development processes as well as service-oriented decision making processes. Transformation of these roles and responsibilities between stakeholders and team members in software development organization is also called the handover of the knowledge, roles and responsibilities. To provide a smooth framework for handover practices in software development organizations, executives and managers prefer the service oriented architecture.

In different scenarios, we can see the importance and involvement of handover practices. In healthcare, nurses may work on different shifts (Morning, Evening and Night etc.) and when a nurse is leaving, he/she is required to handover the condition and report of patients. In a service-oriented scenario, if they work in the distributed environment, they may have an access on their duties. Similarly, in economic transitions, funds are transferred in distributed manner, so that, change on one system takes effect on all connected systems. Similarly, in terms of software sys-

tems when a project is deployed, a user provides their feedback and this feedback is considered as software maintenance task in terms of software engineering. In other words, knowledge is transferring from client side to the development side. In this scenario, we can also see the importance of handover practices and transition of knowledge [1] [2] [3].

In this paper, we focused on different phases of handover process in terms of software engineering as well as service-oriented architecture and its applications for handover process. Primarily, practitioners don't pay much attention to this important process and studies and experience show that sometimes very large mistakes and issues found on deployment side which was not present in test or development environment.

Rest of paper is organized as follow: Section 2 summarize the review of existing literature and extracts the shreds of evidence which motivates researchers to do work on it. Section 3 properly defines and explain the proposed idea, guidelines and section 4 discuss the evaluation of these proposed guidelines and finally, section 5 concludes the research work and discuss the future directions.

## 2. RELATED WORK

Software Development is an activity performed by a different kind of personnel (including business analysts, software developers, testers, etc.). In order to develop and maintenance of any software in any type of organization, it is an intensive activity in terms of stakeholders. At the same time, software developers

### Cite this paper:

Asad Masood Qazi, Mamoona Bashir, "A Service Oriented Handover Process for Software Development Projects", International Journal of Advances in Computer and Electronics Engineering, Vol. 3, No. 5, pp. 1-8, May 2018.

need to communicate with each other, as well as with project managers, requirements engineer, designers and other related service providers. Similarly, when a prototype of software is completed it is important to handover the chunks of software to the clients. In such situations, it is very difficult for developers to manage the complete activities. So, we need to look at the different service providers to strengthen the software engineering activities. The remaining portion of this section describes the importance of service oriented architecture in software engineering in detail.

Consumers may have a different level of requirements. Some of the requirements are concerned with operational level, tactical level and remaining are addressed at a strategic level. In this study, a service oriented approach is developed to build a communication environment between the supplier's environment and customer environment. Requirements are categorized as Critical Priority, High Level Priority, Medium Priority and Low level Priority. At supplier side, SOA is responsible to provide services to customers in terms of gathering requirements and take actions on them using proper and valuable mechanism. In this paper, requirements are assigned to the developer. SOA calculates the effort applied to each requirement and ensures that services are mapped to the service level agreement document. Results of a case study is designed to put proper analysis on quantitative as well as qualitative data. From this research prioritization of requirements are easily managed [4] [5].

It can also be observed from a study given in [6], software development process is a continuous process. A client can give feedback or change request in any form. There should be meta model for each object development and change of the object. The meta model is a service that walks side by side with the development. For the development of any object or any maintenance and change, meta objects have instance of each object. Requests for improvements are generated by the client, and on a response, the development team is responsible to analyze, design, implement and then test the object and handover to the client [7].

It is also important to discuss the challenges in Handover practices, so we can work on them to provide a meaningful solution. There are several gaps and issues identified in different research papers like insufficient system knowledge, lack of domain knowledge, lack of communication, difficulties in tracking the changes and so on [8].

Handover process may have three stages, which are: Pre-delivery (for planning and maintainability), Handover (Handover the data, knowledge, software or hardware) and Post-delivery (evolution and maintenance) activities [9] [10].

During handover process, a checklist is designed before migration of objects. However, if handover

process is not service oriented especially in global software development environment, these handover checklists are not implemented properly. In this study [11] [12] handover checkpoints are designed and properly implemented in a large scale GSD projects of Capgemini SD&M and set as a fundament for implementation and customization of projects.

To conclude the above discussion, it is important to build an automated service-oriented architecture for handover process in software engineering process. If the development team has completed the development tasks, it is important to have a checklist to ensure the minor errors and issues and these should be resolved properly before the objects migration. Secondly, it is important to tackle the software maintenance and change requests properly and keep a record of them. So, decision makers of software development organization can make decisions on it on the basis of client side requests as well as on the basis of the answers and solutions made by a development team.

### 3. PROPOSED WORK

In this research work, we investigate the issues in all three dimensions of Handover process for a software organization. Handover phase is started when the development is completed and objects are ready for deployment in client side environment. Pre Handover phase deals with software quality assurance guidelines and checklists. Handover process is a migration of development objects and post-handover phase is testing and assurance of quality of objects as well as modification requests are generated.

In this section, we will discuss our proposed framework and guidelines in detail. The phases of handover process are presented below in Figure 1.

#### 3.1 Meta model approach for Service Oriented Handover Process

The meta model is actually model about model. In software engineering meta modelling is complete field and provides help to create a domain specific solution by transformation of complex problem into meaningful solution by narrow down the complexities into different possible level of hierarchies. Meta model is also useful to convert model to text (code) and vice versa easily [13] [14]. A meta model for service oriented handover process is properly investigated and presented below in Figure 2.

In propose service oriented architecture for handover process, we deals with the messages between the attributes as well as connection between the attributes and objects. It is as important to keep this knowledge for future use. Knowledge management in service oriented architecture is very important for the decision making [15].

Different studies show that, knowledge management is important for each phase of software



Figure 1 Phases of Handover Process

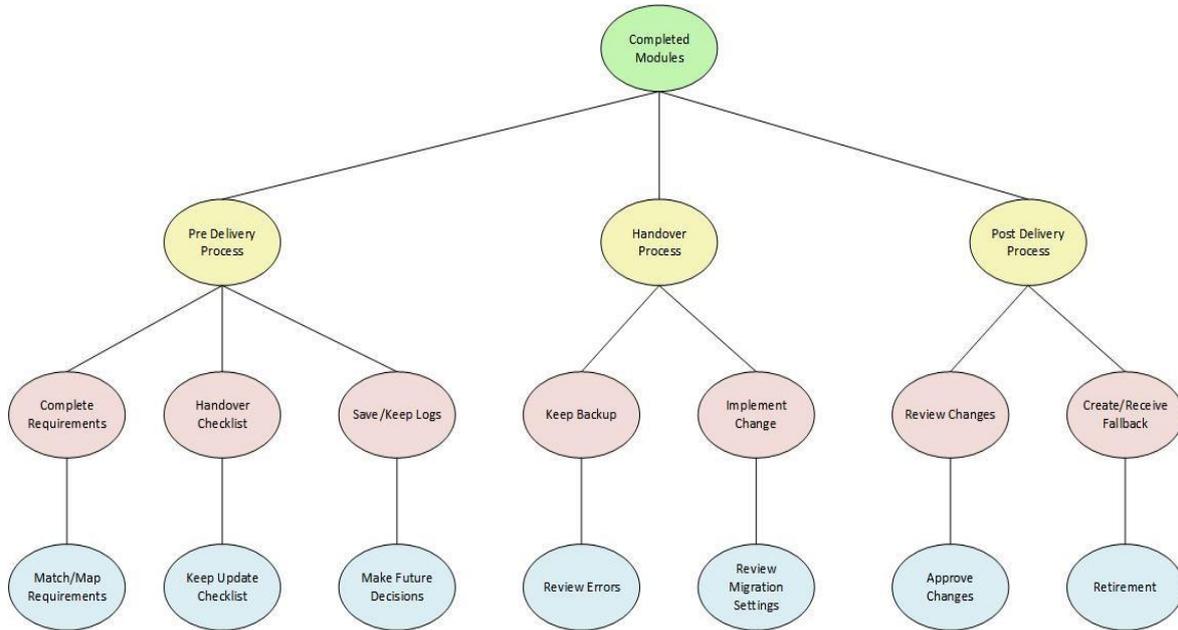


Figure 2 Meta model components of Handover Transition in Software Engineering

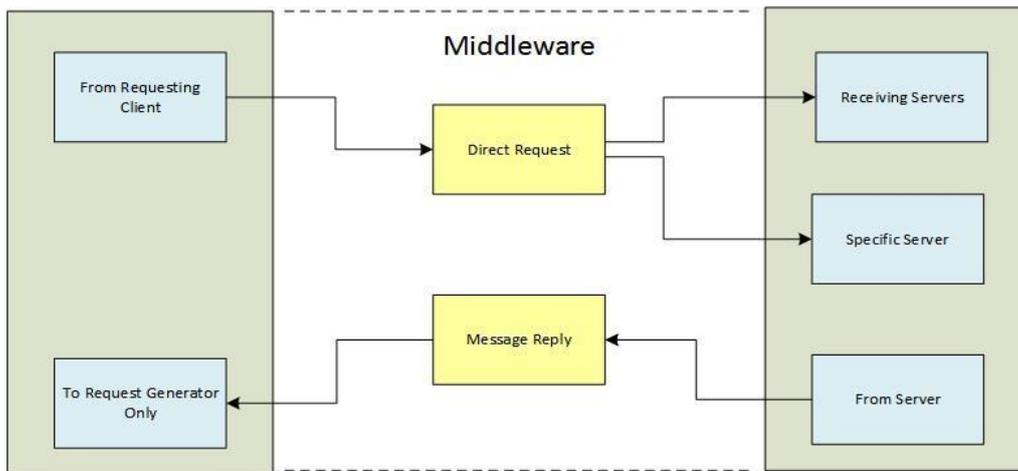


Figure 3 Integration of Middleware Architecture in Service Oriented Handover Process

development life cycle. But unfortunately, knowledge management is not considered as important factor in handover activities. It is important to keep a record of objects migration and handover activities as well as software maintenance data. Once, objects are ready to deployment, it is important to keep a record of that in terms of date, time, number of objects, number of issues, screen shots, log files etc. [16] [17] [18].

### 3.2 Guidelines for each phase of Service Oriented Handover Process

In service – oriented architecture, services could be of any type and could be applied in any phase of the process. It is important to provide the guidelines for each phase of handover process first. In TABLE 1 we have categorized these guidelines for each phase of the Handover process.

TABLE 1 GUIDELINES FOR PHASES OF HANDOVER PROCESS

Pre Delivery Process	Handover Process	Post Delivery Process
Development	Import and Export Development Packages	Testing on Client Side
QA / Check Lists before migration of Objects	Integration of Objects	Validation and Verification of Objects, Data etc.
Accept / Reject Maintenance Requests	Follow up via Email etc.	Generate Requests for Modifications
Pre Delivery Maintenance		Software Support Activities
Post Delivery Maintenance		Retirement
		Software Transition

### 3.3 Stakeholders for each Service Oriented Handover Process

TABLE 2 STAKEHOLDERS FOR PHASES OF HANDOVER PROCESS

Pre Delivery Process	Handover Process	Post Delivery Process
Development Team	Integration Team	QA Team
QA Team	Expert Developer	Client
Requirement Engineer		End User
Project Manager		
Operator		
Maintainer		

It is important to identify the stakeholders for each phase of proposed service oriented handover process for software development projects. The main idea behind the proposed framework is still the same, we intend to make the existing handover process a service oriented. For this, it is important to identify the user to create and transmit the messages / requests and receiver of the messages. The list of stakeholders are presented below in TABLE 2.

## 4. DISCUSSION AND EVALUATION

Service oriented framework for handover practices is properly evaluated in deployment phase of projects in a local software house. In this case study, development chunks and responsibilities are being shifted from development organization to users. The knowledge is being transferred from one stakeholder to another. In below discussion, proper evaluation procedure is discussed. The case study is used to

evaluate the proposed solution and it is followed by the guidelines of case study in software engineering as presented in different studies [19] [20].

### 4.1. Case Description

Proposed framework and guidelines for proper implementation of service oriented handover process is implemented in a live project of a local software house. In this case, a project is related to data warehousing and business intelligence solution. Development team is responsible to develop the required reports, datasets, dashboards and scorecards etc. and later on they deploy the project into client environment.

### 4.2. Demographic Overview

As discussed above, proposed work is implemented in a local software house. The project is related to data warehousing and business intelligence for a healthcare organization. The organization already have prepared their own checklist that is shared with each development team member of the software house. Once, they have completed their tasks, they need to complete the checklist as well.

The main points of checklists are related to the development of objects, format, design, headings, font size, action buttons, links, filters value and their targets and so on. But the issue is, team members often forget to complete the checklist and so that, they don't have any record of maintenance, changes and the items which are deployed previously.

### 4.3. Case Study Design and Development

To overcome such issue in handover process a framework and guidelines are defined in the previous section. Now, it is important to provide the practical usage of these attributes and guidelines. For this purpose, we have design a complete case and implement proposed guidelines in our proposed solution.

Our proposed solution is based upon three phases of handover process and provides following services to the practitioners of software development organization:

**Middleware Architecture:** Middleware architecture is presented between the client and development side to ensure the message passing between the client and development. Figure 3 explain the working of this component graphically.

**Services are distributed:** The distribution of services in terms of message passing is presented below in Figure 4.

**Service oriented checklist:** Checklists contains number of items that each stakeholder must fill during handover process.

**Software maintenance request form:** A sample software maintenance request form is presented below in Figure 5. The data gather from this form is saved in

the database for the post decision making.

**Decision making system for practitioners:** Once data is collected and stored in database, decision making is possible. Each change request is updated into database and analysis could be made according to it. Analysis dashboards are presented below in Figure 6.

#### 4.4. Data collection and Analysis

Once the development and implementation of the proposed solution is completed in a case, it is important to do analysis on resultant data. For this purpose, an interview is conducted as a feedback and the answers are analyzed in terms of satisfactory or unsatisfactory of stakeholders against the framework and proposed solution.

The results show that guidelines for service oriented handover process improve the quality of handover and object migration process. The summary of the analysis is presented below in TABLE 3.

TABLE 3: CASE STUDY RESULTS ANALYSIS

Analysis	Satisfactory Level
Framework is comprehensive	85%
Guidelines of service oriented handover process are well organized	90%
Concept of framework is clear	90%
Reliable for each type of stakeholder	80%
Not required any high level training to implement	90%

#### 4.5. Lessons Learned

From process generation to process execution, each researcher, as well as stakeholders and participants of a case study learn a lot from these activities. The most important lesson that is learned from this research is that each and every phase of software development is important. Handover process is critical because requirements and objects are shifted from one hand to another hand. So, if one person is not able to deliver his or her best in development of project or completion of a task, he or she will not be able to handover properly to other.

Secondly, it is also learned that the success or failure of a project is also be improved using decision making on software maintenance phase. For example, identify the number of requests generated by the user or identify the repetitive tasks of projects in terms of change requests and maintenance.

### 5. CONCLUSION AND FUTURE WORK

Handover process is a critical stage in any software development process especially in large scale projects and global software development environment. In

such environment, the development team may have remote access to deployment server system. It is important to ensure that all the requirements are properly full filled and no ambiguities are there, in deployment package of software.

Secondly, it is also important for the software development organizations to track the maintenance and change requests, generated by the client. And what actions are performed to implement a change, so internally decision making could be possible for any software development organization.

This research work has many future directions. For example proper investigation feature, service oriented in software maintenance area and ensure the message passing between stakeholders in a proper manner.

### REFERENCES

- [1] Berkowitz Daniel, and David DeJong, (2011). "Growth in post-Soviet Russia: A tale of two transitions". *Journal of Economic Behavior & Organization*, Vol (79), Issue (1), pp.133-143.
- [2] Dummer Trevor JB, and Ian Cook, (2008). "Health in China and India: a cross-country comparison in a context of rapid globalization". *Social Science & Medicine*, Vol (67), Issue (4), pp.590-605.
- [3] Savage Christopher, Karlheinz Kautz, and Rodney Clarke, (2017). "A conceptual investigation of maintenance deferral and implementation: Foundation for a maintenance lifecycle model". In *Complexity in Information Systems Development*. pp. 1-16. Springer, Cham.
- [4] Kalinowski, Marcos, Stefan Biffl, Rodrigo Oliveira Spínola, and Sheila Reinehr, (2014). From project-oriented to service-oriented software development: an industrial experience guided by a service reference model. *Journal of Software Engineering Research and Development*, Vol (2), Issue (1), p.10.
- [5] Turner, Mark, Barbara Kitchenham, Pearl Brereton, Stuart Charters, and David Budgen, (2010). Does the technology acceptance model predict actual use? A systematic literature review. *Information and Software Technology*, Vol (52), Issue (5), pp.463-479.
- [6] Krusche, Stephan. and Bernd Bruegge, (2017), May. CSEPM- "A Continuous Software Engineering Process Metamodel. In *Rapid Continuous Software Engineering (RCoSE)*", (2017) IEEE/ACM 3rd International Workshop on (pp. 2-8). IEEE.
- [7] Rodríguez, Pilar, Alireza Haghghatkhah, Lucy Ellen Lwakatara, Susanna Teppola, Tanja Suomalainen, Juho Eskeli, Teemu Karvonen, Pasi Kuvaja, June M. Verner, and Markku Oivo, (2017). "Continuous deployment of software intensive products and services: A systematic mapping study". *Journal of Systems and Software*, 123, pp.263-291.
- [8] Khan, Ahmad Salman, and Mira Kajko-Mattsson, (2010), June. Core handover problems. In *Proceedings of the 11th International Conference on Product Focused Software by ACM*. pp. 135-139.
- [9] Khan, Ahmad Salman, (2013). A Framework for Software System Handover (Doctoral dissertation, KTH Royal Institute of Technology).
- [10] Jabangwe, Ronald, Jürgen Börstler, and Kai Petersen, (2015). "Handover of managerial responsibilities in global software development: a case study of source code evolution and quality". *Software quality journal*, Vol (23), Issue (4), pp.539-566.

- [11] Khan, Ahmad Salman, and Mira Kajko-Mattsson., (2011), June. "Identifying and tackling diversity of management and administration of a handover process". In International Conference on Product Focused Software Process Improvement (pp. 156-170). Springer, Berlin, Heidelberg.
- [12] Salger Frank, (2009), November. "On the use of handover checkpoints to manage the global software development process". In OTM Confederated International Systems" On the Move to Meaningful Internet Systems". (2009). pp. 267-276. Springer, Berlin, Heidelberg.
- [13] De Lara, Juan, and Esther Guerra, (2012), July. "Domain-specific textual meta-modelling languages for model driven engineering". In European Conference on Modelling Foundations and Applications. (2012). pp. 259-274. Springer, Berlin, Heidelberg.
- [14] De Lara, Juan, Esther Guerra, and Jesús Sánchez Cuadrado, (2015). "Model-driven engineering with domain-specific meta-modelling languages". *Software & Systems Modeling*, Volume 14, Issue 1. (2015). pp.429-459.
- [15] Qazi, Asad Masood. and Yasir Hafeez, "Towards Knowledge Management for Predictive Healthcare Business Analytics". *International Journal of Advances in Computer and Electronics Engineering*, Volume 2, Issue 3, (March 2017), pp. 17 – 21.
- [16] Serna, Edgar, Oscar Bachiller, and Alexei Serna, (2017). "Knowledge meaning and management in requirements engineering". *International Journal of Information Management*, Vol (37), Issue (3), pp.155-161.
- [17] Serna, Edgar M., Alexei A. Serna, and Oscar S. Bachiller, (2018). "A framework for knowledge management in requirements engineering". *International Journal of Knowledge Management Studies*, Vol (9), Issue (1), pp.31-50.
- [18] Humayoun, Mamoon and Qazi, Asad Masood., (2015). "Towards knowledge management in RE practices to support software development". *Journal of Software Engineering and Applications*, Vol (8), Issue (08), pp.407-418.
- [19] Tryfona, Catherine, Tom Crick, Ana Calderon, and Simon Thorne, (2017, July). "Software Requirements Engineering in Digital Healthcare: A Case Study of the Diagnosis and Monitoring of Autism Spectrum Disorders in Children in the UK's National Health Service". In International Conference on Digital Human Modeling and Applications in Health, Safety, Ergonomics and Risk Management. pp. 91-98. Springer, Cham.
- [20] Runeson, Per, and Martin Höst., (2009, June). "Tutorial: Case studies in software engineering". In International Conference on Product-Focused Software Process Improvement. pp. 441-442. Springer, Berlin, Heidelberg.

### Authors Biography



**Asad Masood Qazi** is working on Business Intelligence and Data science related projects. Asad also has an experience of research on software engineering principles, software engineering process models and software process improvements. The major work of authors is related to

the implementation of healthcare analytics for a well-known healthcare industry. He has Master's degree in computer science with a specialization in software engineering and also an author of five research articles of a similar domain.



**Mamoon Bashir** is an IT Graduate with a strong background in information technology of telecom sector complemented with Assistant Manager CRM and IT Projects in PTCL. The author is serving with PTCL from four years with depth knowledge and experience of CRM projects and operations, data analysis,

Ad hoc reporting, operational support and analytics domain. Ms. Mamoon is a good team player with strong capabilities of analyzing and resolution of technical issues.

### Cite this paper:

Asad Masood Qazi, Mamoon Bashir, "A Service Oriented Handover Process for Software Development Projects", *International Journal of Advances in Computer and Electronics Engineering*, Vol. 3, No. 5, pp. 1-8, May 2018.

**APPENDIX - I**

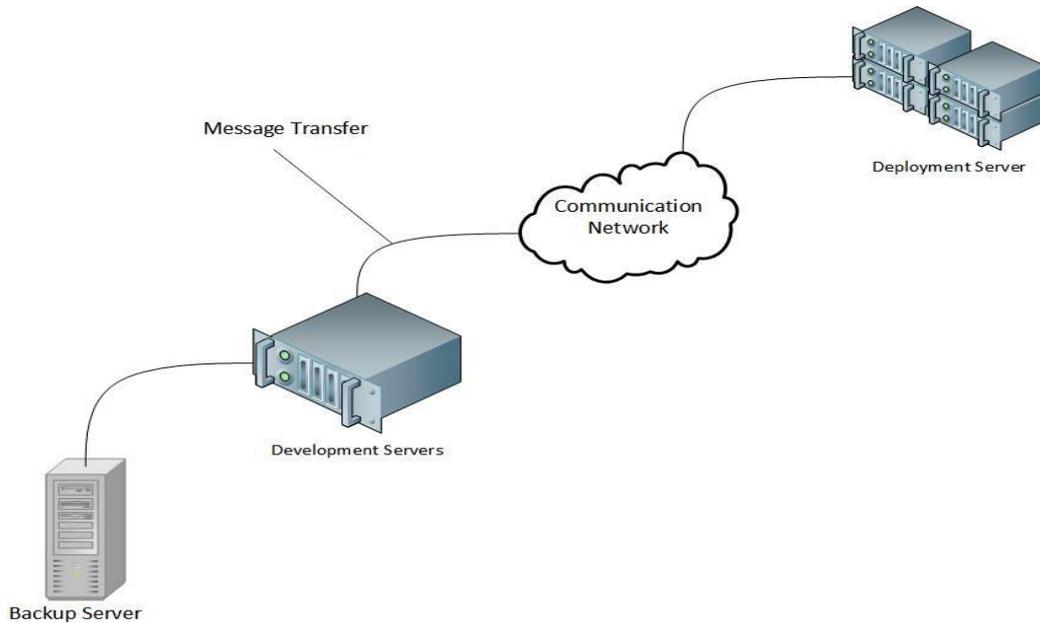


Figure 4 Distribution of Servers for Handover Process in Software Engineering

**Software Maintenance Form**

Please fill out this form as completely as possible. Incomplete forms may be returned for additional information.

Name:	Department:
Email Address:	Phone Number:

This request is to Add  Upgrade  Software

Software Title:	
Version:	Vendor:
License type:	Price per License:
# of Licenses:	Total Cost:
Module:	Services Needed:
Where will the software be installed?	

Who will be the primary user of this software? (Check all that apply)

Executives  Line Managers  Staff

Has this software been fully tested to work in our lab environment? Yes  No

What does this software do and how is it to be used?

How many people a year will benefit from this software?

Please list any additional requirements:

Please submit one form for each piece of software requiring maintenance. You must furnish us with one working copy of the requested software or an address/link where the software can be downloaded.

Please describe your requirements in as much detail as possible. In order to properly implement new technology requests.

Requestor \_\_\_\_\_ Date \_\_\_\_\_

Department Head \_\_\_\_\_ Date \_\_\_\_\_

Approved by Department Head? Yes  No

Vendor Dept. \_\_\_\_\_ Date \_\_\_\_\_

Approved by Vendor? Yes  No

Figure 5 Sample Form for Software Maintenance Request

## Software Maintenance

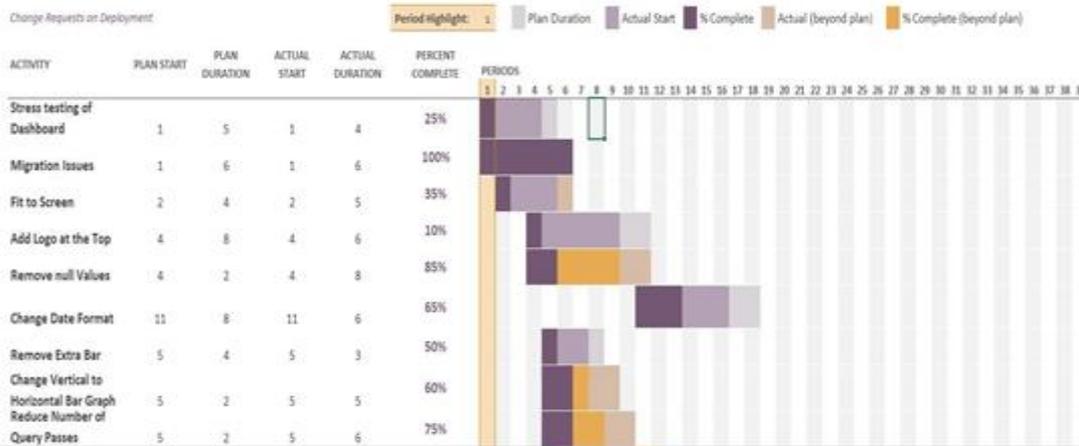


Figure 6 Analysis and Action on Software Maintenance Requests