ST. CHARLES COMMUNITY COLLEGE

CPT-200 - Hybrid - System Analysis and Design

Project Phase Three - Project Management

*Document Prepared by **Rex McKanry** (rmckanry@stchas.edu) and **Dayu Wang** (dwang@stchas.edu)*

Quality Checklist

A **project quality checklist** is a tool used to aid the project team in ensuring they consider all aspects of project and/or process quality. The purpose of well-planned and repeatable quality management is to ensure the delivery of products or services which are acceptable to the customer based on some agreed upon standard of quality. To help achieve consistency many organizations use a standard checklist to verify that all quality considerations have been met during the project planning, execution, and monitoring/controlling phases. Based on the project some checklists may differ, but it is often advantageous to utilize a standard checklist when appropriate to achieve consistency.

Please **click the link below** to download the template of *Quality Checklist*.

CPT-200 - Hybrid - Project Phase Three - Template of Quality Checklist - DOC

Logical Design Documents

1) Use Case Document

The *Use Case Document* is a business document which provides a story of how a system, and its actors, will be utilized to achieve a specific goal. An effective use case should provide a detailed step-by-step description of how the system will be used by its actors to achieve the planned outcome. The purpose of the use case is to tie the business needs of the system to the design parameters of the system to ensure that the completed system achieves the goals established by the business requirements. The level of detail in use cases may vary greatly depending on the size and complexity of the system being designed.

Please **click the link below** to download the template of *Use Case Document*.

CPT-200 - Hybrid - Project Phase Three - Template of Use Case Document - DOC

2) Entity Relationship Diagrams (ERD)

This data modeling technique provides a precise method for detailing and illuminating the interrelationships of the data used by a system. You can depict the "entities" in the data you are modeling and the relationships between them by drawing them onto an entity relationship diagram (ERD). The data model (ERD) shows the major data objects of an application and how they fit together using the relationship. Create an ERD for your system.

Please **click the link below** to access a document that helps you better understand ERD.

CPT-200 - Hybrid - Project Phase Three - Entity Relationship Diagrams - PDF

3) Data Flow Diagram (DFD)

Data Flow Diagrams (DFDs) reveal relationships among and between the various components in a program or system. DFDs are an important technique for modeling a system's high-level detail by showing how input data is transformed to output results through a sequence of functional transformations. DFDs consist of four major components: entities, processes, data stores, and data flows. Create Level **0** and Level **1** DFDs.

Please **click the link below** to access a document that helps you better understand DFD.

CPT-200 - Hybrid - Project Phase Three - Data Flow Diagrams - PDF

Miscellaneous

1) While developers are developing the system, project manager is responsible to communicate with them, asking for any changes of the system, e.g., system architecture change, new services involved, new features added. Please note that any change of the system will **require** changes in related documents, even for those documents already completed in previous phases. Every time when code-document consistency is **broken**, project manager needs to update (or let some one else update) the documents in a timely fashion.
2) Project manager should be the first user of the system. For any progress in development, project manager needs to experience the new added features or other changes. Project manager **cannot** put aside the system development. Even if you do **not** understand code, you still need to try your best to understand the development process. The more you understand about the system, the better you can manage the project.
3) The team **cannot** let the project manager write all the documents. Everyone should contribute to the documents. For example, the developer who designed the system architecture should draw the architecture graph (**not** project manager). It is project manager's responsibility to check and finalize documents.