Hi, Carlie –

Here is a list of tasks that need to be done in order to start on the design and analysis of the new department.

Phase 1 - Planning

* Receive and Approve Project Request
* Conduct Initial Project Meeting
* Feasibility Study
* Create Project Charter
* Formal Approval of Charter

Phase 2 - Analysis

* System and Industry Analysis
  + Evaluate Industry
  + Evaluate user needs/stakeholder needs
    - Questionnaire Development
    - Interviews
    - Questionnaire Results Report
* Prioritize requirements
* Security Risks
* Software Requirement document

Phase 3 - Design

* Data Flow Diagram
* Unified Modeling Language Diagram
* System Design Needs
  + Technician Schedule/Availability
  + Parts Repair Order
  + Follow Up Appointment
  + Calendar
  + Parts Inventory
  + Technician Notepad
  + Service History
  + Mobile Access
  + Inventory requested update every 2 hours to Parts Dept.
  + Customer Discount Calculations
* Report Layout
* User Screen Layout
* User Feedback
* Implementation Plans
* Training Plans
* Maintenance Plans
* Compare Design to requirements
* Prepare Design Document

Phase 4 - Implementation

* Build training environment per design doc
* Testing
* Install System
* Data Conversion
* Documentation
* Train Users

If you feel anything is missing, please let me know. I’ll attach the WBS for your review.

Leah Tritz



Hi, Carlie -

Here are the risks I came up with for the Service Department project. There are some important issues to be prepared for.

**Schedule Risks**

* Inaccurate estimations and scheduling - a poor time estimate is going to cause budget and timeline issues.
* Scope variations - if the scope starts to grow, the timeline and budget will be thrown off and some things will need to be cut.
* Poor scope definition - It’s hard to plan for something if it is not well defined
* Improper resource allocation - too many people working on one part of the project and not on another that also needs to be done will cause inaccurate timelines

**Budget Risks**

* Wrong estimation - underestimating the cost of project tasks will cause a budget overrun.
* Mismanagement of funds - If funds aren’t being spent properly, there will be a budget shortfall.
* Cost overruns
* Improper tracking of budget
* Scope creep - if the scope isn’t well defined, it could inadvertently start to grow which would cause the cost of the project to grow.

**Operational Risks**

* Stakeholder and end user engagement - if they don’t have a say in how it’s set up, they won’t like it. If they don’t like it, they won’t be happy.
* Inadequate human resources - this could be due to high turnover of team members or poorly qualified staff.
* Poor productivity - this can occur if there is high team member turnover.
* Lack of ownership - There isn’t anyone taking responsibility for delivery of software or for the successes and failures.
* Insufficient training
* Compromising on design
* Not testing system with end users
* No support for when the system goes live

**Technical Risks**

* Poor coding quality - if you’re unable to hire qualified programmers, the code won’t be easily used
* Adding extras to the software that are not needed - trying to make it flashy more than functional.
* Frequent requirement changes
* Service outages
* Reduced or obsolete software functionality

**External/Unavoidable Risks**

* Government policy changes
* Rapid market changes
* Fund depletion
* Stakeholders pull approval

I hope this gave you enough to think about over the weekend. Let me know if you need anything else.

Leah Tritz