**Work Breakdown Structure (WBS)**

**Team Two Financial**

**December 2016**

**Table of Contents**

[**Introduction**](#_fyi9lg7he34z)[**3**](#_fyi9lg7he34z)

[**Work Breakdown Structure**](#_b2r8r9iosfxt)[**3**](#_b2r8r9iosfxt)

# **Introduction**

To improve transparency of the multiple projects’ monitoring and execution, an audit firm recommended implementing a Commercial off-the Shelf (COTS) system to be purchased and configured. FinanceMaster software packages were selected because of the modular structure, that is easy to implement to support requirements of the firm without adding unnecessary technological complexity. The entire system can be configured at one time or incrementally. General Ledger/Accounts Payable module is then only one module that is required to be installed.

The Work Breakdown Structure (WBS) selected for the project reflects the high-level tasks needed to gather requirements, implement, and test General Ledger/Accounts Payable module of the FinanceMaster COTS package. The WBS presents logical structure of the tasks that are common to implementation of COTS software and stresses the importance of a structural, balanced approach to high level planning and sequencing of large work packages.

# **Work Breakdown Structure**

**Overview of Project Activities**

1.0 Gather and Analyze Requirements

1.1 Compliance Requirements

1.1.1 Gather HR requirements

1.1.2 Gather IT requirements

1.1.3 Gather Project requirements

1.1.4 Gather Individual user requirements

1.2.5 Gather Regulatory requirements

1.2.6 Analyze requirements

1.2 Technology Requirements

1.2.1 Gather Runtime requirements

1.2.2 Gather Server requirements

1.2.3 Gather Client requirements

1.2.4 Gather Test environment requirements

1.2.5 Analyze requirements

1.3 FinanceMaster requirements

1.3.1 Gather Per-license requirements

1.3.2 Gather Maintenance fee requirements

1.3.3 Gather Training requirements

1.3.4 Analyze requirements

2.0 Design

* 1. Re-review module design approach
     1. Identify conflicts in departmental, user, and project requirements at “compromise” price point module

2.2.1 Utilize SME’s to assist with SI processes

2.3 Identify missing components in proposed “compromise” module group for each:

2.2.1 IT

2.2.2 HR

2.2.3 Projects

2.2.4 End users

2.4 Re-design module groupings to prioritize (x,y,z)

2.5 Build test plan

3.0 Calculate entire cost of project

3.1 Modules to be purchased

3.2 Internal time dedicated to project

3.3 Establish system for tracking cost savings and improvements once implemented.

4.0 Setup new system (without switching over)

4.1 IT sets up internal environment for system transfer

4.2 FinanceMaster sets up software modules

4.3 Develop code (as needed?)

4.4 Develop interface (as needed?)

4.5 Create procedures

4.6 Provide training to relevant staff

4.7 Adjust company processes to reflect new software package

5.0 Testing (without switching over)

5.1 Functional Testing

5.2 End User Training

5.3 User Acceptance Testing

5.4 Create training materials

5.5 Resolve any user issues with software customer support

6.0 Move to production

6.1 Back up all old systems

6.1.2 IT back up

6.1.3 HR back up

6.1.4 Project back up

6.2 Old system shut down

6.2.1 IT system shut down

6.2.2 HR system shut down

6.2.3 Project system shut down

6.3 Start up new system

6.3.1 Check system in IT

6.3.2 Check system in HR

6.3.3 Check system in projects

6.4 Continue to troubleshoot new system

7.0 Documentation

7.1 Develop resiliency documentation

7.2 Create high level design architecture

7.3 Create training materials

7.4 Create troubleshooting KB articles

8.0 Maintenance

8.1 Create maintenance schedule

8.2 Create monitoring controls

8.3 Create periodic auditing schedule