

PROJECT PLAN

CXONE GUIDE

CXGUIDE_PROJECTPLAN.DOC

NOVEMBER 5, 2002

Construx
S O F T W A R E



Advancing the Art and
Science of Commercial
Software Engineering

CONTENTS

1 INTRODUCTION	1
1.1 DELIVERABLE PURPOSE	1
1.2 LIFECYCLE	1
1.3 PROCESS.....	1
1.3.1 Entry Criteria.....	1
1.3.2 Skill Requirements	1
1.3.3 Tools / Techniques	2
1.3.4 Exit Criteria.....	2
1.3.5 Downstream Customers.....	2
1.3.6 Deliverable Combination Options.....	2
2 OVERVIEW	3
2.1 WHO IS PROJECT PLANNING FOR	3
2.1.1 Expert Users	3
2.1.2 Beginning Users.....	3
2.2 HOW TO USE THE MATERIALS.....	3
2.2.1 How to package the Project Plan	3
2.2.2 Redundancy of Information	4
2.2.3 Pointers in the Project Plan.....	4
2.2.4 When to Customize	5
3 PROJECT PLAN CONTENT GUIDE.....	6
3.1 INTRODUCTION.....	6
3.1.1 Overview	6
3.1.2 Deliverables.....	6
3.1.3 Assumptions and Constraints.....	6
3.1.4 Risks and Assets.....	6
3.1.5 Reference Materials.....	6
3.1.6 Definitions and Acronyms.....	7
3.2 MANAGEMENT STRUCTURE	7
3.2.1 Project Lifecycle	7
3.2.2 Project Organization	7
3.2.3 Risk and Asset Management	7
3.2.4 Issue Management	8
3.2.5 Communication.....	8
3.2.6 Startup.....	8
3.2.7 Closeout	8
3.3 PLANNING AND CONTROL	8
3.3.1 Estimate	8
3.3.2 Resource Identification.....	9
3.3.3 Resource Allocation.....	9

3.3.4 Tracking and Control	10
3.4 TECHNICAL PROCESS	10
3.4.1 Engineering.....	11
3.4.2 Technology.....	11
3.4.3 Infrastructure.....	11
3.4.4 Project Artifacts.....	11
3.5 SUPPORTING PLANS.....	11
3.5.1 Configuration Management.....	11
3.5.2 Quality Assurance.....	12
3.5.3 Testing.....	12
3.5.4 Deployment	12
3.5.5 Integration	12
3.5.6 Procurement	12
3.5.7 Operations	12
3.5.8 Staff Development.....	12
3.5.9 Product Acceptance	12
4 LITE PROJECT PLAN CONTENT GUIDE	13
4.1 INTRODUCTION.....	13
4.1.1 Overview	13
4.1.2 Deliverables.....	13
4.1.3 Assumptions and Constraints.....	13
4.1.4 Risks and Assets.....	13
4.2 MANAGEMENT STRUCTURE	13
4.2.1 Project Lifecycle	13
4.2.2 Project Organization	13
4.2.3 Risk and Asset Management	13
4.2.4 Issue Management	13
4.3 PLANNING AND CONTROL	14
4.3.1 Estimate	14
4.3.2 Resource Identification.....	14
4.3.3 Resource Allocation.....	14
4.3.4 Tracking and Control	14
4.4 TECHNICAL PROCESS	14
4.4.1 Environment.....	14
4.4.2 Methods, Tools, and Techniques.....	14
4.5 SUPPORTING PLANS.....	14
4.5.1 Configuration Management.....	14
4.5.2 Quality Assurance.....	14
4.5.3 Testing.....	14
4.5.4 Deployment	15

Copyright Notice

© 2000-2002 Construx Software Builders, Inc. All Rights Reserved.

For further information or support visit www.construx.com.

1 INTRODUCTION

This CxOne user guide discusses how to use *CxTemp_ProjectPlan* and *CxTemp_ProjectPlanLite* to create a project plan deliverable. It provides additional details that complement the comment text in the templates.

See *CxStand_ProjectManagement* for a description of what CxOne requires from a project plan. CxOne's focus is on making sure critical project success factors are addressed. Due to the varied nature of projects CxOne takes into account that the specific content and organization of project plans will need to be flexible and varied to most efficiently address the needs of different projects.

The content of this guide focuses providing background for using CxOne materials to create project plans. Project plans created in this manner can be created efficiently and will provide a strong basis for planning and controlling a project. Where appropriate, this guide provides insights into how CxOne material may be customized and extended to best fit the needs of unique projects.

1.1 Deliverable Purpose

The project plan is the top-level controlling artifact of a project. It defines the managerial and technical processes necessary to deliver the requirements outlined in the project charter.

A project plan may be a single document or a hierarchical collection of artifacts contained under a root document. A project plan may cover a single project, or it may cover a collection of sub-projects.

1.2 Lifecycle

It is normally started as part of the chartering process, and an initial draft may be completed along with the charter. Although the project plan may evolve along with the project, it should always define enough of the project in detail such that upcoming and ongoing work may be efficiently planned and executed. Normally a project plan should define details as far as possible into the future, preferably covering the entire life of a project from the first complete draft.

1.3 Process

1.3.1 Entry Criteria

- Project charter, either approved or in process. If charter is in process, the project plan is presumably being developed iteratively in conjunction with the charter.

1.3.2 Skill Requirements

Engineering management skills are required to produce the project plan.

1.3.3 Tools / Techniques

- *CxTemp_ProjectPlan* and *CxTemp_ProjectPlanLite*
- *CxCheck_ProjectPlan*
- Supporting CxOne materials in the project management sub-CKA.

1.3.4 Exit Criteria

The project plan lives throughout the life of the project, and may experience many iterations. The exit criteria described here should be used for the initial release and significant revisions of the project plan. Parts of the project plan may be released separate from others depending on the nature of the project.

- Completed inspection based on *CxCheck_ProjectPlan*
- If required, signatures on the project plan review and/or approval sections

1.3.5 Downstream Customers

- All project participants and stakeholders
- The project plan drives all project tasks, artifacts, and deliverables.

1.3.6 Deliverable Combination Options

The packaging of the project plan can vary widely depending on the nature and scope of a project. Small projects may have a single project plan document, larger projects may have a hierarchical family of documents that cover information in a project plan.

CxOne templates provide a standard baseline for creating the project plans, but each project should organize and package project plan content to best meet the needs of the project.

2 OVERVIEW

The project plan is the most important artifact on a project, as it defines how the project, and thus all artifacts and outputs, will be created. The project plan represents the definition of how a project will be managed, and is thus the heart of any project management effort.

2.1 Who is Project Planning For

All projects should have a plan that is appropriately scaled to the specific needs of that project. CxOne has distilled knowledge and experience from many sources to create a project plan structure and supporting materials that meet the following goals:

- Pragmatically focus and easy to use
- Scalable across a wide range of project sizes and project types
- Highly tailorable to the specific needs of individual projects
- Supportive of existing industry standards

The project plan materials work in concert with the rest of CxOne to support efficient and high quality planning and execution of software projects, for all ranges of expertise.

2.1.1 Expert Users

CxOne project management materials provide the expert project planner with a set of tools to augment productivity and quality while encouraging standardization. These users will see the CxOne materials as a starting point and adapt using their knowledge and experience to most efficiently meet project needs.

2.1.2 Beginning Users

The materials are also designed to support those with much less project management knowledge and experience. Although background reading and some familiarity with project management is assumed, CxOne project plan materials support straightforward development of project plans that will set the stage for a well managed project.

2.2 How to Use the Materials

Due to the varied nature of software projects, CxOne project planning materials are designed to be highly customizable. There are several things to be aware of when creating a project plan.

2.2.1 How to package the Project Plan

CxOne requires there to be a root project plan artifact, normally a document, for any project. The reasoning behind this is to provide a clear entry point for understanding a project.

As has been noted elsewhere, the packaging of the entire project plan may vary greatly. Normally it will be a collection of artifacts pointed to from the root project plan, including documents, spreadsheets, scheduling and estimation software tool files, other software tool files or databases, etc. i.e. for complex projects the project plan may be little more than an artifact map that describes a family of project planning artifacts.

Project managers will need to use their judgment of how to best package project planning elements. The CxOne materials provide some guidance. A good way to start is to create a single document and break out elements into separate documents based on complexity, workflow, or CM needs.

Where to best allocate content information in a family of project planning artifacts is up to the judgment of the project manager. A balance needs to be struck between avoiding duplicate content that creates CM risks, and creating documents that are easy to use for project participants. This leads into the redundancy and pointer discussions below.

2.2.2 Redundancy of Information

There are several areas of the CxOne project planning materials (and most industry standards) where it appears that the same information is being documented in multiple areas. Project plans can make use of this redundancy in a manner similar to different design views.

For instance, there can be considerable overlap between a project artifact list, the deliverable list, a WBS, and the project plan itself. However, each of these views may be helpful in understanding the project from a different perspective, even if it adds some extra configuration management overhead.

Another case of redundancy is introduced when information is split up between multiple artifacts, and each artifact still retains pieces of the information for that subject. A common example is when a separate QA plan is created for a project. There can be quite a bit of overlap between the QA related elements of the project plan and what a QA plan covers. Redundancy should again be avoided in these cases unless different useful views are being provided. Project leads should decide what information should go where, and then define and enforce that in the CM process. Often in the QA example, the decision is made to put all QA management and process information into the QA plan, and have the project plan contain pointers to the QA plan where the project plan references QA related issues.

On smaller projects, such redundancy is often unnecessary, and the project manager should structure the information in a manner that reduces redundancy while best meeting the needs of the project. This is the major difference between the normal and lite project plan templates. The lite version removes much of the redundancy found in normal template, as multiple views of a simple project are usually unnecessary overhead.

2.2.3 Pointers in the Project Plan

Due to the potential for a family of artifacts and redundancy of information in the project plan, pointers should be used liberally. Pointers allow an easy mechanism to include detailed information by reference. As in the example above, the root project plan can refer to a sepa-

rate detailed QA plan. Often a root project plan will contain many sections that have nothing but a pointer to a more detailed plan.

Pointers can also be used to cross reference information within an artifact. This can be useful when creating overlapping views of information as described in the section above, by explicitly identifying where different views of the overlapping information can be found.

2.2.4 When to Customize

As with all CxOne materials, project planning materials encompass critical issues that should be addressed for most projects. A nominal path is provided by the templates and supporting material that will work well for most projects. To gain greater efficiency the exact organization, content, and presentation can be tailored to best meet project needs.

3 PROJECT PLAN CONTENT GUIDE

This section is the detailed guide for the normal project plan template, *CxTemp_ProjectPlan*. For smaller or less formal projects, consider using *CxTemp_ProjectPlanLite* described in section 4.

3.1 Introduction

Describes any necessary background or context information for the project. Refer to the project charter if one exists; if appropriate, the project's vision statement could be repeated from the project charter.

In some cases it may be appropriate to summarize the project plan itself here.

3.1.1 Overview

This section provides an entry point for understanding the project and the environment it is taking place in. This section may provide more detail on items described in the project charter, but should simply refer to the charter if there is nothing worth calling out in greater detail.

3.1.2 Deliverables

Lists the artifacts that will be delivered by the project to the customer. The level of detail used here should normally be at a summary level, and should mirror any contractual agreements if appropriate.

If appropriate, this section should also include information on the delivery dates, locations, methods, etc.

3.1.3 Assumptions and Constraints

Lists the formal or informal assumptions and constraints that could impact the overall planning of the project. Often the description in the charter is sufficient. Depending on whether separate supporting plans are used, assumptions and constraints may be pushed out to a more specialized plan.

3.1.4 Risks and Assets

Lists the formal or informal risks and assets that could impact project planning. This section will seldom need to elaborate on risks and assets developed during the charter. Risks and assets listed here should identify items that are likely to be a major ongoing factor for project planning. Specific, timely risks should normally be managed through separate extrinsic risk management materials.

3.1.5 Reference Materials

List any documentation referenced by this plan.

3.1.6 Definitions and Acronyms

List any project definitions and acronyms introduced to the project by this plan.

3.2 Management Structure

This section should define the context in which the project is occurring, and how it will be structured to make use of resources, especially staffing. This section is purposely abstracted from the more detailed planning resource management to provide a clear picture of the general ideas being used to plan and execute the project.

3.2.1 Project Lifecycle

This section should describe the complete, customized lifecycle for the project. Normally a well known lifecycle will be selected for a project, so this should be defined and the reasons for selection discussed. Customizations of the lifecycle should be noted, along with any combinations or hybridizations of multiple lifecycles.

This section should also discuss how the lifecycle for the project has been tailored from the standard lifecycle, and describe specific planned instances deriving from that lifecycle, e.g., enumerating planned deliveries for a staged delivery lifecycle.

3.2.2 Project Organization

External Interfaces

Described the important external interfaces for the project. This could include other departments, managers, project reviewers or auditors, etc.

Internal Structure

Describe the structure of the project team.

Roles and Responsibilities

Project roles should be abstractions that define roles to fulfill project activities. Staffing of individuals to project roles should occur separately from defining project roles.

Staffing

The staffing plan is often contained in a separate document if it is expected to be modified frequently. On smaller projects, one person may play many or all roles.

3.2.3 Risk and Asset Management

Depending on how you look at it, all of project management is risk management. The purpose of this section is to describe how risks and assets will be managed on the project.

On many projects it may be appropriate to simply state that risks will be managed intrinsically and through the use of a top risks list in the project status report.

Large, complex, or highly constrained projects may require significantly more formal risks management techniques, e.g., assigning a risk officer.

Relevant CxOne Materials: **CxCheck_RiskLists**, **CxPattern_RisksList**.

3.2.4 Issue Management

This section covers how issues will be managed on a project. Issues is a broad term that can be generic or specific. Issue management is often issued to cover ongoing project issues and do fine grain risk management. Change control and defect management closely resemble issue management and may be merged with the process depending on your needs.

Issue management processes work well when built around database driven tools, but can be managed with a simple list of issues.

3.2.5 Communication

This section should dictate how, when, and what information will flow between project participants and stakeholders. Communication is an intrinsic part of running a sound project and overlaps with other areas of the project plan, especially the tracking area.

Many projects can just call out the major communication devices and processes here, e.g., weekly executive status reports, and leave the details for other sections. Some projects may require a more detailed or formal description of communication needs, or define a schedule of communication artifacts, events, or milestones that need to occur during the project.

3.2.6 Startup

Any significant issues a project needs to deal with to get up and running they should be dealt with here. Many projects in existing organizations can skip this section or refer to organizational processes for starting up projects. This is because startup issues are often implied by staffing and other planning.

3.2.7 Closeout

As with Startup, this section is only needed if closing down the project is a large scale exercise or unique to normal organizational processes.

3.3 Planning and Control

This section should define the planning, tracking, and control mechanisms that will be used on the project.

3.3.1 Estimate

Describe or refer to all estimates to date relating to the project.

Relevant CxOne Materials: **CxCheck_Estimate**

Estimation Process

Describe the estimation process used to create the initial project estimate(s). Outline how and when the estimate will be refined throughout the project.

3.3.2 Resource Identification

List the resources available to the project including information on available staff, time, budget, materials, etc. When appropriate, describe the resource obtainment and phase out timelines.

If appropriate, this section details the gaps between the available resources and the resources necessary to complete the project.

Staff

Describe the staffing on the project. Include planning of the peak staff, staffing profile, and staff build-up of the project. If appropriate, include best case, worst case, and nominal case.

Time

Describe the available calendar time for the project. If appropriate, include best case, worst case, and nominal case.

Cost

Describe the available budget for the project. If appropriate, include best case, worst case, and nominal case.

Materials

List any materials that will be used for the project.

3.3.3 Resource Allocation

Describe the allocation of the resources to meet the project goals. Consider the following:

- Work breakdown structure of the projects activities or artifacts
- Business and detailed schedules for the project activities and tasks
- Budget allocation to different functions during the project (e.g., engineering, quality assurance, management, etc.)
- Staffing profiles throughout the project
- Equipment procurement throughout the project

When appropriate, this section simply contains pointers to other project documentation.

Work Breakdown Structure

The WBS is normally captured in a separate document or specialized tool like *CxTemp_EarnedValueWorkPlan*. The WBS decomposes all project work into work packages. This view allows project work to be managed.

There are two common ways to decompose a WBS, by subsystem and by deliverable release phase. With iterative lifecycles, the subsystem decomposition will contain the multiple deliverable iterations broken down per subsystem. The deliverable release view is orthogonal, with each release forming a top branch that contains deliverables that are part of the release. Depending on the method chosen a WBS may look somewhat like the architecture or schedule; choose a view that best meets your project management needs.

Iteration is often necessary for a WBS, because decomposition using either method can only go so far at early stages of the project and will need to be refined along with requirements and design as the project progresses.

Schedule

Describe or reference the schedule(s) for the project. There are usually several different schedules of varying levels of detail that describe the allocation of work packages. Most schedules are stored in separate documents or specialized schedule tools, but this section may contain a top-level summary schedule.

Budget

Describe the allocation of budget resources to the project. Not all projects require a dedicated budget plan.

3.3.4 Tracking and Control

Describe how project cost, schedule, quality, and functionality will be tracked and controlled throughout the project. Consider describing the following:

- Reports (content, structure, and frequency)
- Audit mechanisms
- Project website
- Time accounting

This section will normally delegate quality tracking to the QA plan and functionality tracking to the CM plan, but impacts of those areas on the top-level schedule should be covered (such as audits or planning checkpoint reviews).

This section and the communication section should take care to compliment each other and not needlessly duplicate information.

Relevant CxOne Materials:

CxCheck_ExecutiveStatusReport, CxTemp_ExecutiveStatusReport

CxCheck_DetailedStatusReport, CxTemp_DetailedStatusReport

3.4 Technical Process

This section should describe the engineering or technical environment in which the project is occurring.

3.4.1 Engineering

Environment

The development environments in which the project will occur. E.g. corporate headquarters, off site, distributed work environment, etc.

Methods, Tools and Techniques

The development methodologies including requirements development practices, design methodologies and notations, programming language, coding standards, documentation standards, system integration procedure, and so on (these will not all be defined when the first draft of the project plan is created; the section should be updated as the plans become more detailed)

3.4.2 Technology

Environment

The computing system environment including hardware and operating system environment;

Methods, Tools, and Techniques

Software tools including design tools, source code control, time accounting, compiler or IDE, debugging aids, defect tracking, and so on

3.4.3 Infrastructure

Discuss any work necessary to create or configure the development environment(s).

3.4.4 Project Artifacts

This section is also known as a document list. An organizational process like CxOne may negate the need for detailed artifact lists if all project artifacts are fairly standard and their delivery is described in other places.

Use this section if there are significant non-standard project artifacts, or there is a reason to capture all project artifacts in one place. This list is generally referenced rather than included here.

3.5 Supporting Plans

This section either incorporates or references the detailed supporting plans necessary to round out project planning.

3.5.1 Configuration Management

Describes how the project identifies and controls the project artifacts including change control, builds, and release processes.

Relevant CxOne Materials:

CxCheck_ConfigurationManagementPlan, CxTemp_ConfigurationManagementPlan.

3.5.2 Quality Assurance

Describes how the project ensures it meets the quality goals including reviews, audits, etc.

Relevant CxOne Materials: *CxCheck_QualityPlan, CxTemp_QualityPlan.*

3.5.3 Testing

Describes how the project will execute and track the testing activities.

Relevant CxOne Materials: *CxCheck_TestPlan, CxTemp_TestPlan.*

3.5.4 Deployment

Describes how project will deploy and support the product.

Relevant CxOne Materials: *CxCheck_DeploymentPlan.*

3.5.5 Integration

Describes integration between sub-projects, tasks, releases, etc.

3.5.6 Procurement

Describes the mechanism for procurement of materials or sub-contracted services on the project.

3.5.7 Operations

Describes the mechanism for ongoing operation of the product.

3.5.8 Staff Development

Describes any significant training, teambuilding, or other staff development activities.

3.5.9 Product Acceptance

Describes the customer acceptance activities for the project.

4 LITE PROJECT PLAN CONTENT GUIDE

This section covers the deltas between the lite project plan template and the normal one.

4.1 Introduction

4.1.1 Overview

See *ProjectPlan* description.

4.1.2 Deliverables

See *ProjectPlan* overview description.

4.1.3 Assumptions and Constraints

See *ProjectPlan* description.

4.1.4 Risks and Assets

See *ProjectPlan* description. Small projects will often describe risks and assets here, but may reference a separate document.

4.2 Management Structure

4.2.1 Project Lifecycle

See *ProjectPlan* description.

4.2.2 Project Organization

See *ProjectPlan* description. May describe the project organization in a less formal manner

4.2.3 Risk and Asset Management

See *ProjectPlan* description.

4.2.4 Issue Management

See *ProjectPlan* description.

4.3 Planning and Control

4.3.1 Estimate

See *ProjectPlan* description.

4.3.2 Resource Identification

See *ProjectPlan* description. May describe resource identification in a less formal manner

4.3.3 Resource Allocation

See *ProjectPlan* description. Small project may describe the resource allocation here via a top level schedule, WBS, etc.

4.3.4 Tracking and Control

See *ProjectPlan* description.

4.4 Technical Process

4.4.1 Environment

See *ProjectPlan* description.

4.4.2 Methods, Tools, and Techniques

See *ProjectPlan* description.

4.5 Supporting Plans

Small project plans will generally include, rather than reference, supporting plans.

4.5.1 Configuration Management

See *ProjectPlan* description.

4.5.2 Quality Assurance

See *ProjectPlan* description.

4.5.3 Testing

See *ProjectPlan* description.

4.5.4 Deployment

See *ProjectPlan* description.