Getting Started with CxOne

Top Level Overview

CxOneOverview.doc

November 5, 2002

Construx
SOFTWARE

cxone

Advancing the Art and
Science of Commercial
Software Engineering

CONTENTS

Copyright Notice

1 Introduction

This document is the starting point for putting **CxOne**[tm] to work on your projects.

Construx's CxOne best practice framework provides a foundation for software project success. You can use CxOne "out of the box" to run projects. Or you can use CxOne as a toolbox, adopting what will add the most value to your projects and organization. CxOne's different versions support variable breadth of coverage and depth of customization to best meet the specific goals of each project.

CxOne is built from the ground up based on the assumption that every project is unique. There is no single right way to run every software project, so CxOne does not prescribe a single way to develop software. Instead CxOne concentrates software industry knowledge, experience, and best practices into a pragmatic and easy to use toolbox.

The philosophy underlying CxOne is further described in *CxOneVision*.

1.1 What is CxOne?

CxOne is an adaptable software engineering process framework. It delivers Construx Software's engineering expertise in a lightweight, scalable, flexible, and tailorable package. CxOne increases productivity of software projects and organizations through pragmatic application of software engineering techniques. It can be adopted wholesale as a project or organizational process, or used piecemeal as a set of software engineering tools.

CxOne is defined primarily through project artifacts. CxOne identifies ***artifacts*** (like a project plan. coding standard, or technology management plan) and provides ***materials*** that support creation of those artifacts (such as checklists, templates, patterns, and guides).

Focusing on artifacts allows CxOne to stay lightweight and grounded in what really needs to be produced for software success. This provides an alternative to traditional process approaches that focus on extensive prescriptive descriptions of how work should be done. CxOne defines knowledge, processes, and tasks, by working backwards from the end goal.

CxOne is based on IEEE's SWEBOK (Software Engineering Body of Knowledge). The SWEBOK defines the universe of software engineering knowledge and is a basis for defining engineering licensing and certification in the software industry.

1.2 CxOne's Scope

CxOne is applicable to most software projects and organizations. CxOne's scope includes:

- Development of software applications and systems

- Development of products and systems containing software

- A wide range of software sizes, domains, technologies, and techniques

- Software engineering concepts, techniques, best practices, and processes

- The full lifecycle of software projects from inception to closeout

- All types of development from custom systems to integration to maintenance, from embedded software to enterprise systems to desk-top and web applications.

CxOne does not require use of specific tools or technologies. It can be paired with existing tools, languages, methodologies, technologies, etc. CxOne does provide support for effectively managing tools and technologies to ensure business goals are efficiently met.

1.2.1 Versions of CxOne

Construx provides four licensed versions of CxOne:

CxOne Basic is available free of charge from Construx's website. CxOne Basic allows for evaluation of a subset of CxOne. See *CxOneBasicEula* for more details.

CxOne Building Blocks provide sets of CxOne materials targeted at specific project needs. Building Block licenses are available for purchase from Construx.

CxOne Toolkits provide full lifecycle support for specific types of projects. Toolkits bring together a set of CxOne materials to support particular project lifecycles, methodologies, or domains. Toolkits are under development by Construx.

CxOne Enterprise is available for deployment through Construx. It provides project and organizational support, a customized range of CxOne materials, and support from Construx consulting and training. Contact Construx for more details.

See the CxOne website, www.construx.com/cxone, for more details on these offerings.

1.2.2 Relationship to Existing Methodologies

CxOne is generally compatible with the many existing processes, methodologies, and techniques available to the software industry. CxOne is not a big-M methodology that seeks to replace what came before it and define how software must be done in all situations. Instead it provides flexible support for getting software done. CxOne can be used alone or to support existing methodologies such as Rational's Unified Process (RUP), Extreme Programming (XP), and design methodologies such as Unified Modeling Language (UML).

CxOne can be used as a toolbox that expert practitioners use to tune their organization by applying targeted improvement to key areas.

For organizations with less mature development processes, CxOne can provide end-to-end software processes based on proven industry best practices. Organizations can use CxOne materials off the rack, or tailor them to their specific needs. In both cases, CxOne is designed to adapt and evolve with the needs of projects and organizations.

1.3 Key Concepts

Critical background and context for understanding CxOne is described here. Browsing *CxOneTermsAndAcronyms* is also a useful way to explore concepts and terminology.

1.3.1 Artifacts and Materials

The term "artifacts" is used to refer to any end product that is necessary to support either a software project or organization. Artifacts are often documents, but include diagrams, web pages, source code, specialized application and database content (e.g., scheduling software), e-mails, software, etc. CxOne relies on the existence of artifacts to implicitly define actions. For example, the existence of a test plan, test cases, and reports of completed tests implies that certain types of testing activities occur. This focus on artifacts allows CxOne to be flexible and lightweight, allowing users to develop processes that best fits their unique situations.

The term "materials" is used to refer to the components of CxOne. CxOne materials support the creation of artifacts. Some materials have a direct one-to-one correlation to particular artifacts. Others provide structure or support for classes of artifacts.

The packaging of CxOne materials into particular artifacts is intended to support easy use of the CxOne resources and map to a reasonable packaging of project artifacts. However, CxOne does not prescribe particular packaging of artifacts, so users should feel comfortable repackaging the content of artifacts to best meet their needs.

1.3.2 Document Tools vs. Software Tools

CxOne is primarily a document based system, which allows software teams to quickly and efficiently address the critical success factors of their projects in a wide variety of environments. Construx believes building effective and efficient software organizations is best addressed through optimizing skills, process, structure, knowledge, and techniques; not by purchasing software silver bullets. We have chosen to build the essential content of our process framework first, and explore automating areas that could benefit from software tools second.

CxOne integrates well with existing software tools, both commercial and internal, allowing them to achieve their full potential. Construx is developing some key software tools that will be part of CxOne in the future, and can create customized deployments of CxOne on request.

1.3.3 Customization

CxOne may be adopted as is, either in whole or in part, to support creation of project artifacts. There is a considerable amount of flexibility built into CxOne, so it is able to support a wide range of artifact creation without the need to modify CxOne materials themselves.

CxOne Enterprise materials can also be customized to optimize for the needs of an organization. Such customization normally involves combining CxOne with existing processes, and then using CxOne's continual process improvement mechanisms to refine and evolve the organization's ability to create software.

1.3.4 Lite Materials

CxOne is designed with extensive scalability and flexibility to support all sizes and types of projects. A key feature of CxOne are "Lite" material versions. CxOne as a whole is light-weight, but smaller or less formal projects may need much less structure than others. CxOne recognizes this fact by providing lite versions of many critical materials that allow for successful, disciplined projects with minimum overhead.

1.3.5 Industry Standards

CxOne provides support for industry standards including IEEE's software standards and Software Engineering Body of Knowledge (SWEBOK), the Project Management Institute's Project Management Body of Knowledge (PMBOK), the Software Engineering Institute's CMM flavors, ISO 15504 (SPICE), and other specialized industry standards.

CxOne draws heavily from Steve McConnell's works and provides a standard for Construx software development, consulting, training, and product offerings.

1.3.6 Existing Resources

CxOne distills the best of existing industry resources and makes then easily accessible for practical use. CxOne does not duplicate or redefine existing knowledge unless it is critical for understanding. Instead the focus is on providing practical tools to apply knowledge, while providing references to knowledge.

CxOne references existing industry literature, standards, terminology, and other publicly available resources including SWEBOK, PMBOK, NASA SEL materials, Steve McConnell's and other author's books and articles, etc.

1.3.7 Self Definition

CxOne materials support efficient high-quality artifact creation on individual projects. They can also be tailored to specific project and organizational needs. In addition, CxOne supports evolution by defining CxOne using CxOne materials. For example, CxOne documents follow the CxOne document standards, CxOne checklists follow the CxOne checklist standards, etc.

Organizations can start with CxOne out of the box and then optimize it over time to best meet their unique challenges.

2 CONSTRUX KNOWLEDGE AREAS

This section provides an overview of the **C**onstrux **K**nowledge **A**reas (CKAs). The CKAs form the basis for organizing CxOne materials.

2.1 Knowledge Area Descriptions

The CxOne CKAs are summarized below. See the top-level standard *CxStand_Xxx* (where *Xxx* is the CKA name) for a detailed description of each area.

Configuration Management

Configuration management (CM) involves revision control, change management, and release management. This includes identifying all elements of a system, organizing and storing artiacts, controlling changes to the system and project, maintaining the integrity and traceability of the system, and creating releasable versions of a system.

Construction

Construction is the application of technology to create executable software according to a design. It includes resolving low-level design issues, defining primitives for data structures and algorithms, creating and integrating content, configuring and integrating third party components, and creating a development environment. The primary activity is creating code to implement functionality using selected languages, technologies, and environments. Also includes coding practices, debugging, support for testing, performance optimization, integration strategies, and construction planning.

Design

The bridge between requirements and construction, design defines the structure and dynamic state of a system using different techniques, different levels of abstraction, and different modeling views. Includes design concepts, practices, techniques, and tools for systems design, architecture, decomposition, data design, distributed and communications design, interface definition, processing models, algorithms, user interface design, physical design, etc.

Maintenance

Activities specific to installation, deployment, maintenance, evolution, migration, and operation of software systems.

Management

Primarily focused on project management, this large CKA is broken into three sub-CKAs.

Project Management is the heart of CxOne engineering management. Includes risk management, estimating, planning, inception and closeout, process and lifecycle selection, scheduling, tracking, issue management, and controlling the outcome of a project.

Organizational Management relates to managing a software organization, managing across projects, and general business management.

External Management covers management of clients, vendors, sub-contractors and other entities external to a project or organization.

Process

Covers defining, measuring, and improving software development processes. This includes definition of software lifecycles and other project and organizational processes. Also includes metrics analysis and control of organizational processes.

Quality

Static activities associated with providing confidence that a software artifact, activity, process, or system conforms or will conform to requirements, standards, and expectations. Includes full life cycle quality planning, quality measurement and analysis, quality management, reviews, audits, verification and validation, root cause analysis, and defect prevention.

Requirements

The requirements CKA includes support for the discovery, analysis, modeling, and documentation of the functions to be implemented in software. This includes concepts, practices, and techniques for eliciting, specifying, controlling, and communicating requirements. Includes support for system analysis that spans beyond software.

Testing

Testing includes dynamic activities associated with executing software to detect defects and evaluate features. Testing includes test planning, test case design and creation, planning and designing automated testing, and techniques for executing specific kinds of tests including bench tests, unit tests, component tests, feature tests, smoke tests, integration tests, regression tests, stress tests, performance tests, acceptance tests, etc.

Tools and Methods

This area of CxOne covers evaluation, selection, and management of tools, technology, and methodologies. Also covers knowledge management and transfer. The use of specific tools, techniques, methods, and technologies to support activities of other CKAs is considered part of the CKA the activity relates to.

2.2 Organization

Organization of CKAs is based on IEEE's Software Engineering Body of Knowledge (SWEBOK) knowledge areas. Construx supports the adoption of SWEBOK by the software industry so has aligned CxOne with the SWEBOK.

Construx knowledge areas expand and modify SWEBOK knowledge areas to support CxOne's needs. For more information about CKAs and their relationship to the SWEBOK, see the Construx paper *Construx Knowledge Areas*.

3 ARTIFACTS AND MATERIALS

Artifacts supported by CxOne have at least one type of material (e.g., a checklist), and often several types of material (e.g., checklist, template, and guide), associated with them. The purpose of the different CxOne material types is described in this section; the packaging of CxOne materials is described in the next section.

3.1 Artifacts

Artifacts are project or organizational deliverables, such as a project plan or source code. CxOne provides *materials* that increase productivity and quality when creating *artifacts*.

The term *artifact* is intended to be somewhat abstract; it may be a document, source code, spreadsheet, tool, or some other deliverable. An artifact may represent a part of or a group of deliverables. In some cases an artifact may refer to a role, responsibility, process, or action instead of a tangible deliverable. Abstract grouping of artifacts in CxOne will not always be 'pure' or 'correct', but should be intuitive and pragmatic.

Artifact usage on a project is tailored to the unique needs of the project. Sometimes there may be a one-to-one mapping with a document that a project would produce, other times a project may split up an artifact into several deliverables or combine several artifacts into one deliverable (e.g., folding CM and quality plans into the project plan on a small project).

3.2 CxOne Materials

Materials support the creation, review, and maintenance of different types of artifacts. The presence of a CxOne material implicitly defines the artifacts it supports (e.g., *CxCheck_ChangeControlPlan* implies the existence of change control plans).

CxOne has ten distinct material types: checklists, templates, standards, guides, patterns, process flows, best practice descriptions, samples, evaluations, and references. Most artifacts will only have a subset of these material types associated with them. The remainder of this section describes the CxOne material types in detail.

Checklists
Checklists define the essential nature of a process, activity, or deliverable. Much of the light-weight knowledge and processes definition that CxOne provides is delivered through checklists. They are intended to be used during creation of an artifact or to review the quality of an artifact. When appropriate checklists may also cover actions, roles, or events. Checklists can also be used to define gates for completion or acceptance.

Templates
Templates are generalized starting points for artifacts. They provide a reusable framework that encourages complete content and conformance to standards. Templates are normally outlines with a small amount of comment text and boilerplate.

Standards

Standards define taxonomy, processes, policies, knowledge, and/or guidelines for creating artifacts, performing actions, or organizing work. Standards may directly define their content, or include references to external information sources. Standards increase efficiency by allowing for reuse by reference. For example a quality plan can refer to *CxStand_ReviewProcess* instead of describing project review processes from scratch.

Guides

Guides provide information on how to use CxOne materials to create artifacts and may also provide educational background.

Patterns

CxOne patterns provide a flexible mechanism for capturing useful concepts or reusable content in a modular fashion. Patterns differ from templates in that templates always refer to a well defined artifact in its entirety, and indicate a strong preference about how to package that artifact. Patterns are more modular, providing ideas or reusable content that can be plugged in where needed. Patterns can also be used to capture ideas, concepts, and reusable material that span several artifacts or which do not map cleanly onto artifacts.

Best Practice Descriptions

CxOne is based on proven industry best practices, so all CxOne materials incorporate best practices in various ways. Best practice descriptions explicitly recognize and provide detailed descriptions of key software engineering techniques, lifecycles, etc. Best practice descriptions provide insight and guidance into how the best practice may be effectively deployed.

Process Flow Handbooks

Process flow handbooks provide directions for bringing together CxOne materials to support particular lifecycles, workflows, methodologies, or other types or processes. Process flows provide a flexible mechanism for customizing and overlaying effective processes on a set of materials to optimize achievement of specific project goals.

Samples

Samples are artifacts from previous efforts that provide for insight, education, and reuse. CxOne comes with illustrative samples, but the real power of samples is unleashed by developing an organizational library of reusable artifact samples. CxOne's framework for cataloging sample artifacts naturally tailors CxOne to the unique needs of that organization.

Evaluations

Evaluations integrate CxOne with external tools, methodologies, techniques, or technologies that may be employed as part of creating an artifact or executing an action or process. CxOne evaluation materials briefly describe the item, define how it can be used, and provide a pointer to supporting material.

References

References are external pointers to related standards, context, knowledge, or training that is outside of CxOne. References incorporate external material and may refer to educational sources that will provide insight or understanding into CxOne or provide information about topics closely related to CxOne. References may be stand alone materials or embedded in other materials such as standards, user guides, best practices, or patterns. Internal references between CxOne materials are handled by links.

3.3 General Issues

3.3.1 Pointers and Links

The term "pointer" is an accepted knowledge management term describing formal reference to knowledge, material, or other content that lies outside of a knowledge framework. This is as opposed to "links" which refer to content within the knowledge framework.

CxOne uses the *CxRef_Xxxx* material type to define **pointers**.

CxOne defines **links** by using the file name of CxOne material in *italics*. To ensure broad tool support, links in CxOne are not yet hyperlinks, but will be in the future.

3.3.2 Hierarchal Organization

CxOne is organized in a hierarchy based on the Construx Knowledge Areas (CKAs). See Section 3 for a description of the CKAs and the organizational philosophy behind them.

3.3.3 Scalability

All of CxOne is designed to scale across size, complexity, quality, and business constraints of different industries, organizations, projects, and products. CxOne material may scale up or down according to the needs of a project or organization.

This scalability is baked into the design of CxOne. Sometimes scalability may be called out more directly in organization or packaging of CxOne content as with lite materials.

4 PACKAGING

This section covers the packaging and conventions used with CxOne materials.

4.1 Material Content

Most CxOne materials are packaged into files that are readily storable, distributable, accessible, and printable. Some CxOne content may be defined through other tools or mechanisms.

4.1.1 File Formats

At this time Microsoft Word and Excel are the standardized document and spreadsheet file formats for CxOne. Some CxOne materials may be packaged as spreadsheets, XML, HTML, databases, or other specialized tools or file types.

For *CxOne Basic*, all materials except templates are distributed as read-only PDF files and web pages. *CxOne Building Block, Toolkit,* and *Enterprise* licenses provide read-write access to native file formats to support tailoring of the CxOne framework.

4.2 CxOne File Conventions

The naming conventions for CxOne materials are described in the table below.

Material Type	Naming Convention	Example
Checklist	**CxCheck_Xxxx**	CxCheck_ProjectPlan.doc
Template	**CxTemp_Xxxx**	CxTemp_Project Plan.doc
Standard	**CxStand_Xxxx**	CxStand_ProjectManagement.doc
Guide	**CxGuide_Xxxx**	CxGuide_ProjectPlan.doc
Pattern	**CxPattern_Xxxx**	CxPattern_ChangeControlProcess.doc
Best Practice	**CxBest_Xxxx**	CxBest_EvolutionaryDelivery.doc
Process Flow	**CxProcess_Xxxx**	CxProcess_WebEvolutionaryLifecycle.doc
Sample	**CxSample_Xxxx**	CxSample_ReviewProcess.doc
Evaluation	**CxEval_Xxxx**	CxEval_UmlClassDiagrams.doc
Reference	**CxRef_Xxxx**	CxRef_ChangeControlProcesses.doc

Note that if you are using *CxOne Basic* many files will be in PDF format.

4.2.1 Lite Content Versions

Lite versions of some content may be packaged separately from normal versions. This is done when the makeup of an artifact for a small project is different from the makeup for large projects. Suffixing the file name with 'Lite' denotes lite versions of materials, e.g.:

CxTemp_ProjectCharter.doc
CxTemp_ProjectCharterLite.doc

4.3 Differences in the Basic and Product Versions

The basic and product versions of CxOne have several packaging differences.

4.3.1 Subset of Materials

CxOne Basic provides a subset of CxOne materials. The complete license of *CxOne Enterprise* includes all CxOne materials. *CxOne Building Blocks* and *CxOne Toolkits* contain focused groups of materials, many of which are not found in *CxOne Basic*.

4.3.2 Read Only vs. Read/Write Content

In *CxOne Basic* all materials except for templates and patterns are provided in read-only formats, usually .pdf files. CxOne products provide all materials in a completely modifiable format.

4.3.3 Distribution

CxOne Basic is downloadable from Construx's website. CxOne products are available on CD in addition to download, and preserve folder organization of CxOne materials.

5 HOW TO START

This section provides assistance with evaluation of CxOne through *CxOne Basic*. It assumes that *CxOne Basic* is being accessed from Construx's public website at:

www.construx.com\cxone.

To begin using CxOne, it is recommended that this overview document be read completely. In addition the following are good reference materials to have handy:

- The *CxOne Basic Document Map* web page.
- The *CxOneTermsAndAcronyms* document.

5.1 Help With a Specific Artifact

If you are looking for assistance with a particular task, such as creating a quality plan, find the CxOne materials that most closely match the artifact(s) you need to create. If materials for the artifact exist, you can use them to assist in creation of the artifact.

If materials do not exist they may not be present in your version of CxOne or there may be a terminology conflict. Please contact Construx at cxone@construx.com for assistance.

5.2 Starting a New Project

If starting a new project or phase of a project, process flows are a useful tool for illustrating how CxOne materials may be used to support your project.

CxProcess_SpsgProject is a great starting point for exploring workflow and processes on a CxOne project. This process flow describes an average project run using the process described in Steve McConnell's Software Project Survival Guide (SPSG).

5.3 CxOne Samples

CxOne comes with some illustrative examples of how CxOne materials can be used to create project artifacts. These can be accessed from the samples area of the CxOne website.