Software Executive Report

Upgrading Your SDLC

Companies seem to be increasingly focused on upgrading their SDLCs (for Software Development Life Cycle or System Development Lifecycle).

An SDLC is the overarching organizational framework that governs how a software project is conducted. At the project level, an SDLC establishes the order in which a project specifies, prototypes, designs, implements, reviews, tests, and performs its other activities. It establishes the criteria that you use to determine whether to proceed from one task to the next.

At higher organizational levels either the SDLC or a coordinated PDLC (product development life cycle or its equivalent) define the major stages and gates a project passes through. The SDLC/PDLC commonly defines the exploratory work needed to support project budgeting, release criteria for the software ultimately developed, and other corporate-level milestones in between.

SDLC Basics

Why have an SDLC? Companies cite several reasons:

◆ Codifying recognized good practices.

◆ Defining enough "hard points" in the development lifecycle to support coordination with all the project stakeholders involved, including technical groups, product support, sales, marketing, finance, and executive management.

◆ "You play how you practice." You should practice desired behavior when you're not under stress because under stress people tend to revert to behavior that's comfortable and familiar, not necessarily to behavior that works best.

The Software Engineering Institute's CMM (Capability Maturity Model) describes five common features that successful process definitions contain:

◆ Commitment to perform

◆ Ability to perform

◆ Activities performed

◆ Measurement and analysis of performance

◆ Verification of performance

These common features essentially make up the elements of a successful SDLC. **Companies tend to focus on the commitment to perform and the activities performed, but in many cases even with commitment and defined activities, their staffs lack the *ability* to perform**—they haven't been properly trained; time to perform the work isn't included in the project schedule; and so on.

Except in regulated environments, companies rarely verify performance, which can lead to a years-long delay in discovering that the commitment to perform isn't translating to actual performance on the ground.

Need for Upgraded SDLCs

Companies report three main reasons for wanting to upgrade their SDLCs:

March ECSE Meetings
Navigating the Planning Cycle

Issues related to long-term planning, planning for individual projects, portfolio management, and budgeting, seem to vary tremendously between organizations. We'll dive into these topics in depth during our March meetings.

Bellevue Meeting: Mar. 10, 5:00-7:00 pm.
Dial-in Meeting: Mar. 14, 8:00-9:00 am, PDT.

Construx®
www.construx.com

Construx®

(Continued from page 1)

◆ A company has never had a well-defined SDLC

◆ A company has a well-defined SDLC, but it originated in computer hardware development, manufacturing, or some other non-software discipline, and it needs to be modified to adapt appropriately to software project needs.

◆ A company has a longstanding SDLC, but it has gotten out of date, calcified, overly bureaucratic, and it no longer expresses how the company wants to conduct its projects

An overarching reason for having an SDLC is that companies concerned about SOX, ISO 9000, FDA, FAA or other kinds of regulations are required to have a defined software life cycle.

Who is Responsible for the SDLC and Who Drives It?

Most companies report that the VP Development, Director of Software Engineering, or similar role owns responsibility for the SDLC.

In most companies, details of the SDLC are driven by a group that reports into the executive level—a PMO (Project Management Office), SQA (Software Quality Assurance function), engineering excellence, corporate platform office, or simi-

lar group. A few companies report that business units drive their SDLC(s).

One company reports that its SDLC is owned by a group that has no official authority but that has high peer recognition of its authority.

> "Companies tend to focus on the commitment to perform and the activities performed, but in many cases even with commitment and defined activities, their staffs lack the _ability_ to perform."

When Can a Project Team Deviate From the SDLC?

Companies report numerous variations in terms of when project teams can deviate from the process defined by the SDLC:

◆ Project teams can make changes if needed to meet their schedules.

◆ Project teams can make changes if needed—but they must document any variations to support ISO 9000 certification.

◆ The project manager has authority to change the process at will—presumption is it's better to ask for forgiveness than for permission

◆ Project manager proposes a project plan at the outset of the project which can include deviations from the normal SDLC; approval of the plan is based on an assessment of whether the PM has the maturity to successfully manage the deviations.

◆ Changes need to be pre-approved by the PMO or equivalent organization.

◆ The SDLC is pretty loose early in the product lifecycle and tightens up as the team approaches product launch.

ECSE Calendar 2008	
March	Navigating the Planning Cycle
April	Managing Core Development
May	Special Issues in Managing Technical Personnel (aka "Guru Management")
June	Balancing "Doing" with "Improving": Improvement Strategies
July	Supporting Innovation
August	_Summer break_
September	Issues in Test Management
October	Compensation Updates
November	Improving Productivity
December	_To be announced_

About the Executive Council for Software Excellence (ECSE)

The ECSE is an executive discussion group hosted by Construx Software. Meeting monthly since 2002, the ECSE's goal is to share, analyze, and evaluate members' experiences facing enterprise-level software development challenges. ECSE members are executives with multi-project, enterprise-level responsibility for software development. The typical member oversees activities of 100 or more software personnel. The ECSE has a few members who oversee smaller staffs.

If you are interested in joining the ECSE or if you know someone who would be interested, please contact the ECSE host, Steve McConnell, at **stevemcc@construx.com** or (866) 296-6300.

(Continued from page 2)

Handling "Compliance" to the SDLC

Companies run the gamut from needing to assure compliance for ISO certification or SOX compliance (and going through formal audits to assure that) to not checking for compliance at all.

Several companies report that their SDLCs are more descriptive than prescriptive—they are essentially collections of practices that work well in 2008. In those cases, only a few practices are mandated, such as security and globalization.

One large company reports that its SDLC is descriptive at the corporate level (i.e., used to provide guidance to business units) and becomes prescriptive when adopted at the business unit level (i.e., prescriptive for individual project teams).

Companies affected by SOX (Sarbanes Oxley) report varying degrees to which SOX affects the details of their SDLCs. The range runs from having a defined SDLC and auditing for general compliance to requiring defined artifacts for nearly every single activity—even on Agile projects.

Construx SDLC-Related Seminars

Construx offers numerous seminars that support SDLC improvement:

◆ 10x Software Engineering

◆ How to be Agile Without Being Extreme

◆ Rapid Development in Depth

◆ Software Project Survival Boot Camp

For more details on Construx seminars, see www.construx.com/seminars/ or contact us at training@construx.com or +1(866) 296-6300.

One company reports having each team explicitly choose the Agile practices it will use from a defined set of practices. The team then fills out a large grid that defines which practices were used, which is seen as sufficient for certification and compliance purposes.

> "Many companies report doing 'Waterfall in the large' while doing 'Agile in the small'."

How Does Agile Fit Into a Defined SDLC?

Companies are wrestling with the question of how to fit agile practices into their SDLCs, while still meeting ISO, FDA, SOX, and other requirements.

What problems does Agile solve best? Agile *construction* increases transparency, which is good. But Agile *requirements* usually decrease predictability, which is not good. In fact, that's a mismatch with most business's reported needs.

This tends to lead companies that have well-defined SDLCs to Waterfallish development models that at least try to get the "vision pillars" or "10,000 foot requirements view" pinned down in the early days of their projects.

Many companies report doing "Waterfall in the large" while doing "Agile in the small." At the 10,000 foot level, the product lifecycle is well defined. At the individual feature team level, feature teams can be agile/iterative in how they go about implementing the work they're responsible for.

About Construx

Since 1996, Construx Software has provided industry-leading support for software development best practices. Through our combination of seminars, consulting, and resources & tools, we have helped hundreds of software-intensive companies better achieve their business goals.

For more information about how we can help your company achieve its business goals, please contact us.

10900 NE 8th Street Suite 1350 Bellevue, WA 98004 +1 (866) 296-6300 ecse@construx.com

Construx®

www.construx.com

The Increasing Role of Scrum in Agile SDLCs

In full-blown Scrum, project teams define which requirements they will deliver on an iteration-by-iteration basis. Since iterations are nominally 30 days in duration, this is problematic for organizations that need predictability longer than 30 days.

A common modification is to define all or nearly all requirements early in the project, treat those as the "product backlog," and then perform the rest of the project using Scrum, including having teams define which specific requirements they will implement in each 30-day iteration. In other words, the agile practices become more about sequencing than about requirements flexibility per se. This approach provides many of the technical and management benefits of short iterations while also supporting predictability of the eventual project outcome.

> "We still try to get most of our requirements up front, especially the overarching vision pillars."

Issues in Scaling an SDLC

As organizations scale up, their SDLCs tend to become more detailed and more prescriptive. Most staff see the value of such changes. However, organizations commonly report that a few longstanding staff members ultimately leave because "everything is becoming too structured"—or they have to be asked to leave be-cause they're not willing to work within the SDLC.

Editor's Note: SDLCs are a Friend to Better Software Practices

The last few years have seen an interesting development: companies concerned about SOX are becoming more serious about their software development practices. They realize that longstanding (but not very effective) cowboy practices that provide poor project transparency and predictability are a corporate liability. Software projects that cost millions of dollars need to be managed with the same standard of care as other kinds of projects that cost that much.

In this context, companies are increasingly establishing PMOs or their equivalents to provide governance over their software projects. The PMOs in turn are insisting that project teams use practices generally recognized for producing good results.

This has the potential to move the software industry in a healthy direction. Keys to success include making sure that the PMO is staffed with experts who are respected by their technical peers; ensuring that the SDLC the PMO defines matches the business's needs; ensuring that any detailed technical practices defined by the PMO are a good cultural fit for the organization; and providing enough training and mentoring to ensure that the management and technical staff have the ability to execute whatever practices have been defined.

— Steve McConnell

SDLC Support

Construx provides support for SDLC improvement including SDLC Definition, SDLC Review, SDLC Facilitated Workshop, Agile Practices Review, Transitioning to Agile Practices, Audits, and Report Cards. In addition, Construx provides cradle to grave training in software development best practices to ensure that technical staff has the ability to perform the practices defined in the SDLC. For information about how Construx can help your organization, please contact us at consulting@construx.com or +1 (866) 296-6300.

Construx Software, 10900 NE 8th Street, Suite 1350, Bellevue, WA 98004 +1 (866)296-6300