Construx®

2017 Catalog

Professional Development—for Software Professionals

Training the Way You Want It

On Site
at your location

OnDemand
online learning

Blended Learning
the best of both worlds

Public Seminars
at our Seattle/Eastside training center

Agile Development
Project Management
Software Requirements
Design & Construction
Software Testing & QA
Software Estimation

Contents

Choose your learning mode(s)

Contact us today, and let us customize a training program that will best meet your specific training, scheduling and budgetary requirements.

➤ **hello@construx.com | +1 (866) 296-6300 | www.construx.com**

On Site Onsite Training

Construx has helped hundreds of companies with the highest impact classroom training in the software industry.

Choose Onsite training to:

- Provide a shared experience for an intact team
- Maximize learning and immediately put new skills into practice
- Facilitate rich interactions among learners
- Receive live feedback from Construx's expert instructors
- Reward and inspire your team with the highest quality software training experience available

On Demand OnDemand Learning

Building on 20 years' experience delivering classroom training, Construx now offers industry-leading OnDemand Learning.

Choose online, OnDemand Learning to:

- Provide maximum training convenience for your team
- Enable team members to learn at their own pace, and at the times and locations of their choice
- Deliver consistent training to a geographically distributed team
- Reward your team with the best online training in the software industry, while maximizing use of your training dollars

Blended Learning Blended Learning

Experience the most innovative and impactful training in the software industry via Construx's Blended Learning.

Choose a blended combination of classroom training and OnDemand Learning to:

- Deliver a backbone of software practice training to your entire staff, cost-effectively
- Strategically provide high-impact classroom training to the people and teams who will benefit the most
- Realize the best possible combination of training impact and training value, tailored by Construx to your specific training needs

Public Public Seminars

Software professionals say that Construx public seminars are an exceptional use of their time and deliver a great return on their training investment.

Choose public seminars to:

- Focus on training—away from office interruptions
- Learn from peers who are facing the same challenges you are
- Try a seminar individually, before bringing the seminar in for your whole team
- Accelerate your learning in a focused environment

Construx OnDemand Learning

On Demand

10X Software Engineering

Scrum Boot Camp

Code Complete Essentials

Agile Developer Boot Camp

Requirements Boot Camp

Software Economics Boot Camp

API Design and RESTful Services

Product Owner Boot Camp

Agile Requirements

Total Project Quality

Design Patterns

Risk Management In Depth

ALL ACCESS PASS

The All Access Pass gives you a year of access to every software development online offering in the Construx OnDemand catalog, including many new offerings that will be added over the next year. That's a value of more than $5000!

One year All Access Pass for $599.

GET A FREE 30-MINUTE TRIAL AT

cxlearn.com

Steve McConnell's
Understanding Software Projects

LEARNING SCENARIO #1

Blended Learning

Cost Effectively Train a Geographically Distributed Team

Scenario

Your company has grown by acquiring small companies. Now you have a staff of 100 technical personnel spread across numerous sites with 1-15 software professionals per site. You can't justify sending an instructor to each location, and the travel cost of bringing all team members together at one site for in-person training is prohibitive.

Blended Learning Solution

Construx delivers the same high quality OnDemand training to your staff at every location. This avoids travel costs and builds a shared understanding and shared vocabulary through a consistent training experience for team members across the company.

To reinforce the OnDemand training, Construx delivers expert, in-person training at your central office location to a select group of leaders and other staff. These individuals receive in-depth training so that they can perform effectively as thought leaders and coaches.

LEARNING SCENARIO #2

Blended Learning

On-Boarding New Staff Members

Scenario

Your company provides training periodically, and you want new staff members to come up to speed between in-person training sessions.

Blended Learning Solution

When you purchase in-person training courses from Construx, you'll also get full access to the OnDemand version of the course for additional employees. You can use the OnDemand course to provide training to team members who were unable to attend the in-person training. Employees who join the team after the in-person training occurred can use OnDemand training to get up to speed with their colleagues. Every team member receives the same high quality learning content regardless of when they take the OnDemand training.

Customized Training Solutions

Contact us today, and let Construx customize a training solution for you that optimizes your staff's professional development and delivers lasting value for your training investment.

➤ **hello@construx.com | +1 (866) 296-6300 | www.construx.com**

LEARNING SCENARIO #3

Blended Learning

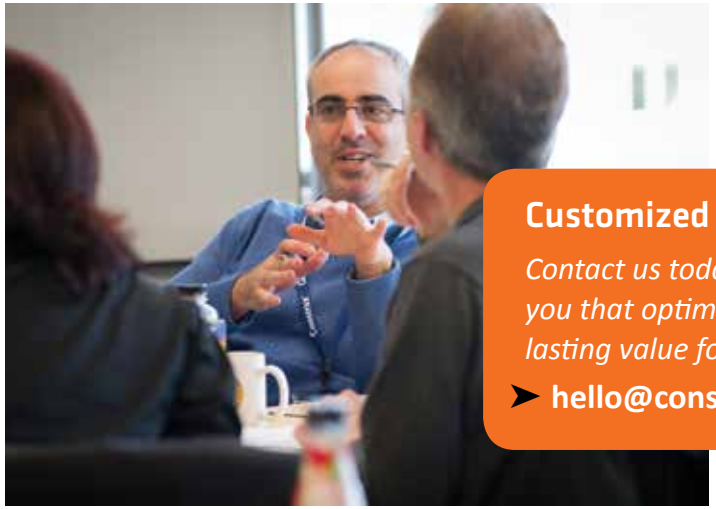
Customized In-Person Training to Address Your Specific Challenges

Scenario

Your company is committed to professional development, and you want to realize maximum benefit for the time your staff spends in classroom training.

Blended Learning Solution

In-person training is scheduled with one of Construx's subject matter experts. Prior to the live training, your staff has access to the OnDemand version of the course, which they consume before the in-person class. This enables the staff to begin the in-person training sharing a solid understanding of the course content. The in-person classroom time can now focus more intently on company-specific implementation issues yielding the most significant results.



LEARNING SCENARIO #4

On Demand

Reward Your Staff and Benefit Your Company with High Quality Training

Scenario

Your company wants to provide professional development for your staff, but day-to-day project demands always seem to take precedence over training.

On Demand Training Solution

You purchase **All Access Passes** to Construx OnDemand training for your entire technical staff. This allows every team member to take any of Construx's OnDemand training modules at any time. If three team members have a few hours of downtime on a Friday afternoon, they can get together in a conference room and consume a few OnDemand training modules related to their current project challenges.

Technical staff views training as a reward—and an affirmation that the company has a stake in their professional development. Construx's state-of-the-art OnDemand training helps your staff feel valued, and your company benefits from their enhanced skills and morale.

Customized Training Solutions

Contact us today, and let Construx customize a training solution for you that optimizes your staff's professional development and delivers lasting value for your training investment.

➤ **hello@construx.com | +1 (866) 296-6300 | www.construx.com**

CASE STUDY #1

On Site

Scaling Agile Across the Enterprise

Bolstered by its success with small-scale Scrum projects, a Fortune 500 computer hardware company hoped that it could transform the organization by simply replicating its small-project successes across the enterprise.

If only it were that simple.

After a reality check, the company enlisted Construx to develop a customized Scrum Boot Camp seminar that was piloted, refined, and delivered to the 2500-person staff at sites worldwide. Construx supported the rollout with executive briefings, onsite coaching, and continuous fine-tuning. The result was a consistent, efficient and fully integrated adoption of Scrum throughout the enterprise.

CASE STUDY #2

Blended Learning

Taking the Team to the Next Level: Success with Larger Projects

A medium-sized scientific instruments company consistently experienced success with project teams of five to nine members. But projects with teams of 15-100+ struggled or failed outright.

Construx developed a customized training program that focused on the unique challenges of succeeding with Scrum at scale.

Training throughout the company included team- and management- level best practices that have proven successful for larger projects. Our experienced Agile experts also provided customized coaching and mentoring, and helped the staff round out their skill sets with On-Demand training in planning, estimation, requirements, and quality assurance.

CASE STUDY #3

Blended Learning

Customized Agile Training for a Small Team

A small company in a narrow vertical market needed to improve its skills in Agile requirements, in performing the Scrum Master and Product Owner roles, and in the basics of Scrum. Their challenges included having a total staff size of nine people and a modest training budget.

Construx delivered a customized blended program that included:

- Three OnDemand courses that team members could consume in sections as needed: Scrum Boot Camp, Agile Requirements, and Product Owner Boot Camp
- An onsite, in-person coaching workshop to fine tune the on-the-ground practices and team interactions
- Follow-up support and coaching via phone and Skype

The blending of OnDemand and personal onsite training was especially effective for helping the company identify and address specific issues, integrate lasting improvements into the team's dynamics, and achieve maximum effectiveness with its Scrum and Agile practices.

CASE STUDY #4

Blended Learning

Expanding Software Development Capabilities at a Traditional Manufacturing Company

The CEO of a traditional manufacturing company launched an initiative to upgrade the company's software capabilities in order to deliver key functionality in its hardware products. Construx was selected as the training and consulting partner with the experience to work effectively with both software and hardware staff.

The company had identified software estimation and software project management as its most significant areas of weakness. Construx helped the company refine its software estimation and project management practices, and align them with the stage-gate development process the company used for both hardware and software.

Construx then delivered an in-person training program in software estimation and project management throughout the company. The training was targeted primarily to software staff, and many hardware staff members attended the training voluntarily.

CASE STUDY #5

Blended Learning

Taking a company from "Scrum, but..." to "Scrum brilliant"

A global professional services company underwent an objective assessment of its software practices and found that its Scrum implementation was mostly "Scrum, but..." In order to achieve its critical software development objectives, the company prioritized improving staff skills in Scrum and Requirements.

Construx was enlisted to develop a customized, company-wide program in Scrum and Agile requirements that included:

- In-person workshops at multiple sites
- OnDemand courses
- Personal follow-up coaching by Skype and email

The OnDemand courses were primarily intended for on-boarding new staff members who had missed the in-person workshops. They also gave the whole staff a convenient way to review content and refresh the skills developed in the workshops.

CASE STUDY #6

On Site

Agile Transformation in a Regulated Industry

A company working under FDA regulation wanted to adopt Scrum but was uncertain how to do so without risking non-compliance with rigorous FDA requirements.

Construx provided customized Scrum training to the team that had been designated to pilot Scrum on an FDA-compliant project. After a highly successful pilot, Construx supported an incremental Scrum rollout strategy by providing training and coaching to additional teams throughout the organization. Internal stakeholder satisfaction soared, and the company made a transformational migration to Scrum while consistently meeting all FDA regulatory requirements.

Scaling Scrum

On Site | On Demand | Blended Learning | Public

NEW Scrum has been a successful way for individual teams to deliver value quickly. As the needs of the business expand beyond what individual teams can accomplish, how can organizations scale Scrum while remaining agile? How do you scale Scrum to multi-team projects, multiple geographies, and work that spans multiple iterations? This two-day course presents tactics used to scale software projects, focusing on techniques that have shown themselves best suited for scaling Scrum. Using an extended case study, you will scale Scrum from a single, focused team to a multi-team, multi-national, distributed project. You will examine scaling tactics from a Scrum perspective including Lean, Kanban, and Scrum of Scrums. You will learn what has worked and what has not worked in scaling Scrum, gaining insights from Construx's extensive experience with clients around the world as well as from other attendees. **[14 PDUs]**

Scaling Fundamentals
- Product vision's critical contribution to scaling
- Infrastructure and tool sets for scaling
- Project focus and multi-tasking - how to manage for scaling
- Patterns of scaled work distribution

Agile Scaling Principles
- Building on software projects' intellectual phases
- Defining clear decision lines
- Setting scope of authority
- Creating collaboration
- Delivering customer facing value consistently
- Maintaining feedback loops
- Living with the hype-disillusionment learning curve

Scrum as a Building Block for Scaling
- Three roles
- Three artifacts
- Four meetings
- Two levels of commitment

Lean Scaling Principles
- Limiting waste
- Early defect detection
- Acknowledging variation

Kanban
- Setting up a pull system
- Creating a work queue
- Setting work in process limits
- Defining classes of service
- Calculating flow time

Technology and Scaling
- Traditional scaling tools
- Web 2.0 support for scaling
- Using instant messaging effectively
- Leveraging shared workspaces
- Electronic meetings—making them efficient and effective
- Configuration management techniques for scaled environments
- Build challenges in scaled environments

Architecture's Role in Scaling
- Using common architecture successfully
- Conway's law
- Architecting for features vs. components
- Planning to allow the architecture to emerge
- Test-driven architecture
- Handling work in Scrum that does not create user-visible value
- Addressing work that spans multiple iterations

Team Coordination
- Planning for time together
- Aligning plans and processes
- Creating convergence points
- Defining clear roles and responsibilities
- Building and using common infrastructure

...and more

See complete course description at **www.construx.com/learn**

Scrum Boot Camp

On Site | On Demand | Blended Learning | Public

UPDATED This seminar provides everything you need to know to ensure your transition to Scrum is successful. In the years since the Agile Manifesto, Scrum has emerged as the most popular Agile process for managing software development projects. More companies are switching to Scrum, but many are struggling. This three-day seminar combines key aspects from Certified Scrum Master- and Certified Scrum Product Owner-specific training plus specific best practices based upon the instructor's direct experience on Scrum transitions and experience gained from Construx's consulting engagements. **[21 PDUs]**

What is Scrum?
- Agile origins, principles, and benefits
- A brief history of Scrum
- Scrum philosophy and theory
- What makes Scrum different?

Why Scrum Works
- A simple process
- A committed, self-managed team
- Transparency: nowhere to hide
- Finishing what you start
- Continual improvement

Scrum Roles
- The Scrum Master
- The Product Owner
- The Team Member
- Stakeholders
- Levels of commitment

Scrum Processes and Meetings
- Release planning
- Sprint planning
- Daily standup
- Sprint review
- Sprint retrospective

Scrum Artifacts
- Product backlog
- Sprint backlog
- Release burndown
- Sprint burndown

Scrum Best Practices
- Timeboxing: Nothing concentrates the mind...
- Commitment: Either do or do not; there is no try
- Working agreements: This is how we do it
- Acceptance criteria and the definition of "done"
- There is no 'I' in team

Spinning Up Scrum
- Selecting the Product Owner
- Creating a product backlog
- Setting up the Scrum team
- Planning the release
- Launching the first sprint

Life During Sprint Time
- A day in the life of a Scrum team
- A day in the life of a Scrum Master
- A day in the life of a Product Owner

Tracking Progress: Scrum Metrics
- Whiteboards or software?
- How is your iteration going?
- Exposing and removing impediments
- When are you going to release?
- Are you improving?

Something's Rotten: Scrum Smells
- Scrum (Task) Masters
- The Product Dictator
- The tyranny of the urgent
- They're just not that into it: when the team fails to meet commitments
- Self-unmanaged teams
- Just give me the fish!
- When burndowns don't burn down
- The plague known as 'Scrum-But'

Learning to Fly
- Scrum is simple but not easy
- Best practices are still applicable

...and more

See complete course description at **www.construx.com/learn**

Scrum Product Owner Boot Camp

On Site · On Demand · Blended Learning · Public

UPDATED The Product Owner role is arguably the most important role in Scrum—and the most challenging. The Product Owner is part project manager, part product manager, and part customer advocate. This person must ensure the customer's wants and needs are understood while ensuring that the team delivers the greatest-value features as quickly as possible—all while responding to ever-changing requirements. Ultimately, the Product Owner owns a Scrum project's success or failure. Scrum Product Owner Boot Camp drills down into the detailed information needed to successfully plan releases, reflect stakeholder priorities, ensure the team builds the right product, and communicate with marketing, sales, executives, and other project stakeholders. **[21 PDUs]**

It All Starts With a Vision
- The Product Vision

The Product Owner's View of Scrum
- What Is Scrum?
- How Scrum works
- Scrum roles, processes and artifacts

The Successful Product Owner
- Technical…enough
- Product Owner, Project Manager, or Product Manager?
- Attributes of successful Product Owners

What Are We Trying To Build?
- What is a requirement?
- The product vision as the top-level requirement
- Three purposes of requirements
- Using requirements to manage risk
- Product versus project requirements
- User stories and acceptance criteria
- The definition of "done" and why it matters

How Will We Build It?
- Successful Scrum projects require planning
- The product backlog
- Creating a product roadmap
- Release planning
- User story mapping
- Incremental delivery via Sashimi implementation

During the Game
- Product backlog grooming
- Sprint planning
- Sprint reviews and retrospectives
- Dealing with change
- Scrum project tracking

See complete course description at **www.construx.com/learn**

> **"Very impressed with your company and team. I look forward to getting more of my team members in your training."**
>
> *Don Montgomery, FIS*

Kanban Software Management: Using Lean Methods and Practices

On Site · On Demand · Blended Learning · Public

Today's global high tech industry abounds with smart people yet many organizations continue to struggle to deliver projects on time, on budget, and with high quality. While people and technology remain part of the equation, much can be gained from a systematic view of processes. Kanban provides an objective, quantifiable approach to managing and measuring the flow of work across an organization. Harnessing concepts from Lean/TPS, Goldratt's Theory of Constraints, Deming's Theory of Profound Knowledge, and Drucker's theories on management, you'll discover practical methods to streamline your organization. [7 PDUs]

What Problem Are We Trying to Solve?
- The goal: delivering value with predictability, efficiency, and quality
- The problem: as projects get larger, efficiency and quality decrease non-linearly, for reasons we don't understand

Changing Our Thinking
- How most organizations approach their work
- False assumptions based on conventional wisdom
- Utilization versus productivity
- The hidden impact of waiting

Deming's System of Profound Knowledge
- Appreciation for a system
- Knowledge of variation
- Theory of Knowledge
- Psychology
- PDSA - Deming's application of the Scientific Method to management

Lean Thinking
- Understanding value
- Identifying waste in all of its forms
- Understanding how our products are built
- The concept of flow
- Pull versus Push
- Achieving excellence via the pursuit of perfection
- LAMDA - Lean Thinking's PDSA-based value-creation cycle

Theory of Constraints
- What is The Theory of Constraints?
- Defining 'constraint'
- The Five Focusing Steps - Theory of Constraints' improvement cycle

What Is Lean-Kanban?
- The Origins of Lean-Kanban
- Lean Principles
- The Kanban Method

Why Kanban?
- What problem are we trying to solve?
- How does Kanban help us identify our problems?
- How does Kanban help us solve our problems?

Understanding Kanban Core Principles
- Start with what you do now
- Initially, respect roles, responsibilities, and processes
- Agree to pursue evolutionary change
- Encourage acts of leadership at all levels

Applying Kanban Practices
- Visualize
- Limit WIP
- Manage Flow
- Make process, policies, and assumptions explicit
- Implement feedback loops
- Improve collaboratively, evolve experimentally

Implementing Kanban in Your Organization
- Building Your Kanban
- Operating the Kanban

See complete course description at **www.construx.com/learn**

Agile Developer Boot Camp

On Site | On Demand | Blended Learning | Public

This seminar provides hands-on experience in proven engineering practices to help software developers work effectively in a team using any of the leading Agile development approaches. Participants will learn how to deliver high-quality, valuable, working software frequently while accommodating changing requirements. As leading Agile frameworks encourage developers to be more generalists than specialists, this seminar goes beyond just coding practices, providing participants with skills that will leverage the twelve Agile principles. Note: Agile Developer Boot Camp focuses on Agile engineering practices. Scrum/Agile processes are covered in our Scrum Boot Camp seminar.

Agile Development Foundations
- Agile values and principles
- Scrum, XP, and Lean software development
- Agile development teams

Automated Testing
- Adding value with automated tests
- Test automation frameworks
- xUnit test patterns

Build Automation & Continuous Integration
- Automated build scripting
- Continuous integration infrastructure
- Essential practices for continuous integration

Version Control Strategies
- Using version control
- Workspace management
- Branching strategies

Collaborative Development
- Interfacing with other roles
- Pair Programming
- Collective code ownership

Test-Driven Development
- TDD cycle and its rules
- Acceptance Test-Driven Development (ATDD)
- Keeping test code clean

Clean Coding
- Coding standards
- Code analysis
- Defining "done"

Individual Productivity
- Getting things done the Agile way
- Pomodoro technique
- Productivity principles

Refactoring
- Emergent Design
- Tracking and paying off technical debt
- System metaphors

Components & Dependency Management
- Keeping application releasable
- Dividing code base into components
- Managing dependencies

Requirements Refinement
- Refining and breaking up user stories
- Removing ambiguities
- Behavior-Driven Development (BDD)

Developer Testing Best Practices
- Ensuring functional coverage
- Domain testing
- Structural testing

Story & Task-Level Estimation
- Story points estimation
- Decomposition and recomposition
- Structured individual expert judgment

See complete course description at **www.construx.com/learn**

Scrum Essentials for Leaders

On Site | On Demand | Blended Learning | Public

This one-day seminar provides a comprehensive overview of Scrum to individuals that need to support Scrum teams and the organization in its transition to Scrum. You will obtain a good understanding of how the work is estimated, planned, and executed in Scrum projects, which will enable you to contribute more effectively in the success of the projects you are involved with. You will also learn key strategies to align and define governance with Scrum projects, and to form and maintain self-organizing, cross-functional development teams. This course in not intended for individuals who are performing any of the three Scrum roles. For these individuals, we recommend our Scrum Boot Camp seminar.

Why Agile
- Why companies are going Agile
- Delivering value early
- The Agile Manifesto

The Scrum Workflow
- Why companies use Scrum
- Scrum workflow
- Scrum roles, events, artifacts, and rules

Scrum Roles
- Scrum Master
- Product Owner
- Development Team

Forming Cross-Functional, Self-Organizing Teams
- Stages of competency
- Cross-functionality matrix
- Raising levels of delegation and authority
- Communities of practice

Product Planning
- How software requirements are handled differently in Scrum
- Populating the initial Product Backlog
- User stories and epics
- Early prioritization

Release Planning
- Flipping the Iron Triangle
- Initial backlog refinement
- User story mapping

Relative Sizing and Velocity
- Accuracy vs. precision
- Planning Poker
- Deriving release content and duration

Sprint Planning
- Definition of "Ready" and Definition of "Done"
- Task decomposition
- The Sprint Backlog

Sprint Execution and Project Tracking
- The Daily Scrum
- Sprint burndown chart
- Release burndown chart

See complete course description at **www.construx.com/learn**

> **"I found the Scrum training incredibly valuable for me and my team."**
>
> *Metin Gokdemir, Baker Hughes*

Software Estimation in Depth for Agile Projects

On Site | On Demand | Blended Learning | Public

Agile development and Scrum in particular have opened the door to powerful new estimation approaches. This course provides many useful rules of thumb, procedures, and lightweight math for creating software estimates ("the art of estimation") and briefly introduces mathematically-intensive approaches to creating software project estimates ("the science of estimation"). This course features extensive lab work to give you hands-on experience creating many different kinds of software estimates--for large, medium, and small projects as well as calibrating estimates to be accurate for your specific development environment. This seminar is based on the best selling book *Software Estimation: Demystifying the Black Art*, by Steve McConnell. Two days. **[14 PDUs]**

Estimation Background
- Estimation "art" vs. estimation "science"
- Estimates, targets, and commitments
- Estimation focus: features, schedule, cost/resources
- Kinds of estimates: macro vs. micro, top-down vs. bottom up, algorithmic vs. heuristic
- State of the art and potential for estimation accuracy

Estimation Process
- Differences in estimating agile projects vs. plan-driven projects
- Basic steps in creating a software estimate
- Best estimation approaches by project phase
- Estimate refinement
- Standardized estimating procedures for agile projects

Estimation Error
- Software's Cone of Uncertainty
- Applying the Cone of Uncertainty to Agile projects
- Building your own Cone of Uncertainty
- The good, the bad, and the ugly: evaluating estimates
- Measuring Error: Residual Error, RE, MRE, VFE, and BRE
- Detecting errors in the estimation process itself

Popular Estimation Methods
- Off-the-cuff estimation
- Typical judgment-based estimation
- Structured expert judgment
- Decomposition
- Group estimation techniques
- Estimation checklists

Better Estimation Methods
- Counting, computing, and judgment
- Story points
- Planning poker
- Velocity
- Release burn-down charts
- Iteration burn-down charts
- Wide-band Delphi
- Estimation by analogy
- Proxy-based estimation
- The PERT formula
- Improving individual contributor estimates

Special Issues in Feature/Scope Estimation
- The Feature Staircase
- The Feature Cone of Uncertainty
- T-Shirt Sizing
- Software's diseconomy of scale

See complete course description at **www.construx.com/learn**

Agile Planning and Estimation

On Site | On Demand | Blended Learning | Public

Learn state-of-the-art methods and practices for planning, estimating, and managing Agile projects. The real-world practices presented in this seminar are based upon evolved concepts and support both basic Scrum projects and Agile at scale—and they have been proven on the ground through extensive use by Construx's clients. These concepts are illustrated throughout the seminar with numerous hands-on exercises. After the seminar, attendees will understand and be able to effectively plan and estimate their projects in days instead of weeks, and be able to confidently forecast project completions with accuracy. **[14 PDUs]**

Agile Project Planning Strategies
- The Agile Paradigm Shift: No More Planning Around Activities That Don't Matter
- The Purpose of a Project
- What do We Mean by Planning and Estimating?
- Deliverables-based Planning
- The Minimum Viable Product
- Object-Oriented Planning and Design by Contract

Envisioning: The Foundation for Planning
- Begin With The End in Mind: The Product Vision
- Who Owns the Vision?
- Creating an Effective Product Vision: The Elevator Test

Planning by Focusing on Deliverables
- User-centered Functional Design
- Prioritizing Users and Features

Effective Agile Work Decomposition
- Agile Work Breakdown Structures
- Creating the Product Backlog
- Hierarchical Story Mapping: Prioritizing the Product Backlog

Agile Estimation
- The Purpose of Estimation
- Why Estimation Is Hard, and Why It Doesn't Have to Be
- Agile Estimation Concepts
- Story Point Estimation Principles
- Playing (and Winning) the Estimation Game
- Agile Estimation Practices
- Estimation Practices for Large Projects
- Effective Organizational Estimation Practices

Release Planning
- Backlog Sequencing Strategies
- Release Planning Principles
- Project Staffing and Budgeting
- Creating the Project Schedule
- Avoiding "Schedule Chicken"
- Understanding Scope, Team, and Project Velocity

Backlog Refinement
- What is Backlog Refinement (Grooming)?
- When and How to Groom
- The Definition of Ready

Effective Sprint Planning
- Using Velocity as a Guideline, Not a Straightjacket
- Planning the Sprint
- Sprint Planning Smells
- Estimation During Sprint Planning
- A Discussion on Commitment

Component Task Creation and Estimation During Sprints
- What is a Task?
- Dynamic and Static Task Decomposition Approaches
- Rules for Sizing Component Tasks
- Lightweight Task Estimation Practices

See complete course description at **www.construx.com/learn**

Software Estimation in Depth

On Site · On Demand · Blended Learning · Public

UPDATED This seminar provides many useful rules of thumb and procedures for creating software estimates ("the art of estimation") and briefly introduces mathematical approaches to creating software project estimates ("the science of estimation"). This seminar features extensive lab work to give you hands-on experience creating many different kinds of software estimates—for large, medium, and small projects—as well as calibrating estimates to be accurate for your specific development environment. [14 PDUs]

Estimation Background
- Estimation art vs. estimation science
- Estimates, targets, and commitments
- Kinds of estimates: macro vs. micro, top-down vs. bottom-up, algorithmic vs. heuristic
- State of the art and limits on estimation accuracy
- Surprise: Estimation's real role on software projects

Estimation Process
- Basic steps in creating a software estimate
- Estimating agile projects vs. estimating plan-driven projects
- Best estimation approaches by project phase
- Estimate refinement
- Standardized estimating procedures for agile and plan-driven projects

Estimation Error
- Evaluating estimates: the good, the bad, and the ugly
- Errors in the estimation process
- Sources of project uncertainty
- Software's Cone of Uncertainty

Popular Estimation Methods
- Off-the-cuff estimation
- Using expert judgment successfully
- Wide-band Delphi
- Cocomo II

Better Estimation Methods
- Estimation by analogy
- Decomposition
- Proxy-based estimation
- Estimation by function points
- The PERT formula
- Putnam's Method

Special Issues in Scope Estimation
- Counting, computing, and judgment
- Fuzzy Logic
- T-Shirt Sizing
- Software's diseconomy of scale

Special Issues in Effort Estimation
- Productivity variations across types of software
- Calibration
- Industry data
- Historical company data
- Project data
- Estimating individuals' work

Special Issues in Schedule Estimation
- The schedule equation
- Effect of schedule compression and expansion
- The impossible zone

Automated Estimation Support
- Product demos
- Tool capabilities
- Interplay of the art and science of software estimation

Human Roles in Estimation
- Estimate presentation techniques
- How to explain and defend an estimate
- Estimation and negotiation

See complete course description at **www.construx.com/learn**

Software Project Management Boot Camp

On Site · On Demand · Blended Learning · Public

Leading any project can be a challenge. Leading a software project can be even more challenging if you're new to project management or new to software. This seminar will help you make the transition to solid software project leadership. Software Project Management Boot Camp teaches you the concepts and techniques necessary to manage projects successfully. This seminar closely follows the Project Management Institute's (PMI) Project Management Body of Knowledge (PM-BOK) and shows how to apply these best practices to a typical small-to-medium sized software project. This course involves extensive hands-on practice with real-world case studies. [21 PDUs]

Introduction
- Defining software project success
- Understanding the challenges on a software project
- Typical software project outcomes
- Construx's path to software project success

Basic Survival Concepts
- Understanding labor rate, burn rate, capital vs. expense
- Taking advantage of the upstream/downstream effect
- Recognizing the intellectual phases of a software project
- Fundamentals of software project estimation

Project Initiation
- Chartering the project
- Assessing risks to software project success
- Recognizing software project assets

Project Close Out
- Typical close-out tasks
- Using a project retrospective to learn from the experience

Project Planning
- Using a software project plan template
- Developing work breakdown structures (WBS)
- Simplifying the WBS with a project matrix
- Building the WBS dictionary
- Choosing a project organization
- Choosing a project lifecycle
- Typical software project effort allocations
- Creating an activity network (PERT chart)
- Finding the critical path
- Developing a realistic project schedule
- Scheduling to fixed end dates
- Addressing uncertainty using rolling wave planning
- Tuning the plan to the specifics of your project

Execution, Checking, and Correcting: Succeeding in Stages
- Controlling change
- Using earned value to objectively track project status
- Conducting effective status meetings
- Creating useful project status reports
- Refining the project plan based on actual progress
- Capturing valuable project history in a project log
- Sanity checking the project using planning checkpoint reviews

See complete course description at **www.construx.com/learn**

Increasing Software Innovation

On Site | On Demand | Blended Learning | Public

Much is written about the "secrets of innovation". Why then are so few companies innovating? What's missing? This seminar will challenge your thinking about everything you've ever heard about innovation. It provides a predictive model that explains why some companies innovate and others don't — a model to help you avoid wasting time and effort trying "secrets of innovation" that won't work for your organization.

You'll leave this seminar knowing how to benchmark your organization's ability to innovate and implement innovation practices specific to your organization. 3-day seminar. [18 PDUs]

Great Products
- Identify characteristics of great products, and learn what made them great
- Understand the impacts of wrong decisions at the product or release definition stages
- Recognize the limitations of the traditional product introduction process
- Apply a practical working definition for great products
- Focus innovation on highest stakeholder value
- Understand the impact of technology evolution on product development

Effective Product Teams
- Understand the differences between effective and ineffective teams
- Overcome five characteristics of dysfunctional product teams:
  - Friction
  - Avoiding individual responsibility
  - Evading measurement
  - Not innovating
  - Not winning
- Increase team motivation
- Break down the walls and work as a team

Product Core Teams
- Structure innovative teams
- Foster business and technical collaboration
- Replicate the startup environment
- Create entrepreneurial teams with the Product Introduction Core Team Maturity Model
- Develop future leaders

Designing Innovative Products
- Establish a product vision
- Apply the five steps of 360° Product Introduction:
  - Quantify stakeholder value to focus innovation
  - Bridge business and technology with user scenarios
  - Define features that maximize stakeholder value
  - Prioritize features based on business value and development cost
  - Create requirements that support innovation
- Leverage 360° Product Introduction with Scrum
- Incorporate elements of great design

Planning Great Products
- Incorporate two important life cycles into your planning:
  - Technology life cycle
  - Technology adoption life cycle
- Apply a collaborative Phase Gate process that leverages technical innovation

See complete course description at **www.construx.com/learn**

Risk Management In Depth

On Site | On Demand | Blended Learning | Public

The project was a guaranteed success—until the subcontractor announced a three-week delay and your chief architect quit to go hiking in Nepal. If you don't attack project risks, they will attack you! Learn intermediate and advanced strategies you can use on both general and project-specific risks. Discover how to identify, address, and eliminate sources of risk before they grow into major problems. This two-day seminar focuses on intermediate and advanced strategies you can use to manage general risks and details practical techniques you can use to control your project's specific risks. [14 PDUs]

Introduction
- A definition of risk
- What is risk management?
- The need for risk management

Risks in Detail
- The scope of risks
- Risk as cause-effect
- Ultimate causes and ultimate effects
- Risk timeframes
- Assets

Risk Identification
- Categories of risks
- Common project risks
- Practical techniques to identify risks and assets

Risk Analysis/Prioritization
- Risk probabilities
- Risk severities
- Techniques for accurately estimating risk probabilities and severities
- Determining risk exposure
- Prioritizing risks
- Analyzing/prioritizing assets

Risk Response Planning
- Risk response strategies
- Prevent
- Mitigate
- Transfer/share
- Contingency plan
- Risk reserve/provision Ignore
- Planning risk response

Risk Responses
- Responses to requirements problems
- Responses to inadequate project management
- Responses to inattention to upstream quality
- Responses to misunderstood project dynamics
- Strategies for maximizing common assets

Risk Response Control
- Response control through project tracking
- Ongoing risk reassessment

See complete course description at **www.construx.com/learn**

Software Economics Boot Camp

On Site | On Demand | Blended Learning | Public

Companies are in business to maximize the return on their investment. Unfortunately, most software professionals lack the skills to evaluate the business impacts of the technical choices they make – which can lead to waste in time, money, and personnel. Grounded in engineering economics and business decision making, this 2-day seminar teaches you the essential concepts and techniques that will help you make technical decisions that positively impact the all-important bottom line. This seminar is taught by Steve Tockey and is based on his book, *Return on Software*. **[14 PDUs]**

Introduction and Foundations
- Business on purpose
- Harsh realities
- Seminar goals
- Roadmap

Fundamentals of Business Decisions
- Business decision-making process
- Selection criteria
- Proposal
- Cash-flow instance
- Cash-flow stream
- Cash-flow diagram
- Developing cash-flow streams

Interest: the Time Value of Money
- Time is money
- Interest
- Naming conventions in interest formulas
  - Simple interest
  - Compound interest
  - Compound interest formulas

Comparing Cash-Flow Streams
- Simple comparison of two proposals
- Equivalence
- Bases for comparison
  - Present worth
  - Future worth
  - Annual equivalent
  - Internal rate of return
  - (Discounted) payback period
- Project balance

Developing Mutually Exclusive Alternatives
- Independent proposals
- Dependent proposals
  - Co-dependent proposals
  - Mutual exclusive proposals
  - Contingent proposals
- Developing mutually exclusive alternatives
- "Do-nothing" alternative
- Cash-flow streams for alternatives

For-profit Decision Making
- Minimum Attractive Rate of Return
- Basic for-profit decision process
- Example for-profit decision
- Rank on rate of return

Allowing for Inaccuracy in Estimates
- Knowledge drives accuracy
- Common ways to allow for inaccuracy
  - Increase MARR
  - Shorten planning horizon
- Better ways to allow for inaccuracy:
  - Use ranges of estimates
  - Sensitivity analysis
  - Delay final decisions

Multiple Attribute Decisions
- Introducing multiple attribute decisions
- Different kinds of "value"
- Measurement scales

See complete course description at **www.construx.com/learn**

> **"I wish my entire company could attend! Nearly overwhelmed by all the new knowledge I can share."**
>
> *Whitney Yadrich, DEG Digital*

> **"Everything about the training was great – the content, the pace, and the instructor."**
>
> *Rinette Scarso, Senior Services Project Manager, Microsoft*

Agile Requirements In Depth

On Site · On Demand · Blended Learning · Public

Agile development shifts traditional requirements work to a "just in time" approach. How does this affect good requirements practices? This seminar explains Agile approaches to traditional requirements sources including MRDs, PRDs, feature lists, and user scenarios. It dives into techniques for developing requirements on Agile projects, including the Agile Work Breakdown Structure (WBS), using story mapping to define the scope of the project, writing user stories, sizing stories (agile estimation), and developing acceptance criteria for user stories. Concepts are illustrated through extensive use of hands-on labs. [14 PDUs]

What Are We Trying To Build?
- It all starts with the product vision
- Characteristics of good product visions

Software Requirements: What and Why
- What is a requirement?
- The product vision as the top-level requirement
- Three purposes of requirements
- Using requirements to manage risk
- Product versus project requirements
- Potentially useful requirements artifacts
- Working software as the ultimate requirements specification

Initial requirements gathering
- Envisioning the high level requirements
- What can I do with it: high-level user stories and story mapping
- How it works: business rules and the domain model
- How it looks: low-fidelity UI models and workflows

Just-in-time Requirements Elaboration
- No requirement before its time: the concept of the last responsible moment
- Requirements elaboration during iteration planning

Requirements Change Management

Requirements Validation
- Acceptance criteria
- The definition of "done" and why it matters

See complete course description at **www.construx.com/learn**

Agile Requirements Modeling

On Site · On Demand · Blended Learning · Public

This seminar gives you hands-on experience using five basic requirements models to more efficiently and effectively elicit and analyze functional requirements. You'll create Context Diagrams, Activity Models (workflow models), Domain Models (E-R models, class models, data models), Use Cases and State Models. You'll gain proficiency at using these models in practical and Agile ways to more precisely and concisely capture requirements without getting caught up in modeling semantics. You'll see how these five models can enable you to gather more requirements earlier in the project, and why model-based requirements exhibit greater stability than those that are interview-based. The seminar will also sharpen your instincts for knowing when you've done enough requirements work to proceed, and where requirements risks remain. **[4 PDUs]**

Define Our Model Toolkit
- Define the model
- Requirements modeling toolbox

Model Your Context
- State the boundary
- Identify actors
- Classify into sets
- Name the data flows
- Prioritize

Define Use Cases
- Define tasks
- Select the primary actor
- Confirm value proposition
- Develop post-conditions
- Develop pre-conditions
- Write a description
- Select a normal course
- Separate actor and system steps
- Create alternate courses
- Define exception courses
- Add specific information

Create Activity Models
- Examine the activities
- Start from the end
- Align actors and sub-tasks
- Sequence the activities
- Show decision points
- Fork parallel activities
- Sync activities
- Identify technology artifacts

Start a Class Model
- Find the nouns
- Search other models
- Examine data cohesion
- Name the classes
- List important attributes
- Run the pit test
- Look for data coupling
- Associate the classes
- Assign cardinality

Utilize State Models
- Find "status"
- Describe a state
- Borrow from activity models
- Bring an instance into existence
- Transition to a new state
- Record transition rules
- Delete an instance

Find the Requirements

See complete course description at **www.construx.com/learn**

> **"The different techniques for prioritizing requirements, the samples and templates are some of the best I've seen in over 20 years in the industry."**
>
> *Don Evatt, Cisco*

Model-Based Requirements

On Site | On Demand | Blended Learning | Public

This three-day seminar provides in-depth, hands-on coverage of developing and documenting model-based functional requirements as part of a Model-Driven Architecture (MDA) software development approach. The software requirements are documented using a subset of the Unified Modeling Language (UML) instead of in a Natural Language like English. Model-based requirements are more complete, concise, less ambiguous than Natural Language requirements and lead to developing higher quality software with less cost and shorter schedules. No prior knowledge of object-oriented development, UML, or MDA is required. The companion seminar, Object-Oriented Design, explains how to complete the MDA process and create code that satisfies these requirements. **[4 PDUs]**

Introduction
- Why model?
- Software methods and models
- Introduction to UML

Fundamental Principles
- Importance of complexity
- Abstraction
- Cohesion and coupling
- Design to invariants

Analysis vs. Design
- Requirements vs. non-requirements
- Functional requirements vs. non-functional requirements
- How to separate
- Why to separate
- Definitions of analysis and design

Object Orientation
- Concepts and terminology of object-orientation

Use Case Modeling
- Actors
- Use cases
- Participates
- Includes & extends
- Generalizations

Class Modeling
- Classes
- Attributes
- Domains
- Associations
- Multiplicity
- Generalization

Interaction Modeling
- Sequence
- Communication

State Modeling
- States
- Events
- Transitions
- Actions

Summary of OOA Modeling
- Cross-model consistency
- Stimulating models

Scaling Up
- Recursive design/domain separation
- Decomposing domains into subsystems

Overview of Designing from an OOA

See complete course description at
www.construx.com/learn

Requirements Boot Camp

On Site | On Demand | Blended Learning | Public

What is the most frequently reported cause of software project failure–regardless of project size or type of software? Requirements challenges. Discover how leading-edge companies use requirements engineering to support successful software projects. Learn the three purposes of requirements and how to distinguish between requirements fantasies and requirements reality. Practice practical techniques for exploring user needs, capturing requirements, controlling changes, and building highly satisfactory software. [21 PDUs]

Software Requirements: What and Why
- Requirements: fantasies and real world
- What is a requirement?
- Three purposes of requirements
- Product and project requirements
- Levels and types of requirements
- Characteristics of good individual requirements
- Characteristics of good sets of requirements
- The vision statement as the top-level requirement
- Requirements as a risk management activity
- Knowing when you're done

The Requirements Process
- Comprehensive strategies for defining requirements
- Iterative elicitation, analysis, specification, and validation
- Breadth-first approaches
- Depth-first approaches
- Spiral approaches
- Tools: chartering workshop, collaborative development, risk management, parallel development

Requirements Elicitation
- Who has requirements?
- Eliciting requirements from people
- Eliciting requirements from other systems
- Eliciting requirements from the environment
- Finding the decision maker
- Incorporating business rules
- Dealing with ambiguity
- Tools: interviews, context-free questioning, brainstorming, JAD workshops, prototyping, task analysis, use cases, competitive benchmarking, document archeology, project charter, vision statement

Who Defines Requirements
- The requirements engineer
- Requirements engineering roles
- Skills needed to develop requirements effectively
- How the requirements engineer relates to the rest of the project
- Checklist for requirements leads

Requirements Analysis
- Classification and prioritization schemes
- Requirements negotiation
- Tools: prototypes, use cases, essential systems modeling, data dictionary

Requirements Specification
- Characteristics of a good requirement specification
- Models as specification

See complete course description at
www.construx.com/learn

Web API Design and RESTful Web Services

On Site | On Demand | Blended Learning | Public

The REST architectural style defines a set of principles and constraints that promote system scalability, loose coupling, reduced latency, security, and the encapsulation of legacy systems. While REST is relatively easy to understand, it does not provide any guidance on how to create programmable "Web APIs". This course will help attendees acquire a thorough understanding of how to design RESTful Web APIs that support CRUD, transactional scenarios, workflow, and even complex systems integration.

Fundamental Web API Design

- Review of HTTP
- The 3 web service API Styles
- Why use Resource-Oriented Web APIs?
- When not to use?
- REST Recap
- Resource APIs and service contracts
  - Logical components of the contract
  - The debate over Service Descriptors
- APIs for CRUD and other more complex scenarios
- Client/Service Interaction patterns
  - Request/Response
  - Request/Acknowledge
- Communicating business logic errors to clients

Behind the Web API

- Service Controllers and data-binding
- Implementation patterns
  - Transaction Script
  - Operation Script
  - Command Invoker
  - Data Source Adapter
  - Workflow Connector
- Request and Response Mappers

Advanced Web API Design

- Client/Service Interaction patterns
  - Long Poll
  - Media Type Negotiation
- Patterns for Complex Searches
  - Query strings
  - Simple Search DTOs
  - Query Tree DTOs
- Transactions and transaction management
- Workflow
- Service composition
- Hypermedia and the Linked Service pattern

Service Evolution, Versioning, and Governance

- What changes in Web APIs cause them to break clients?
- API version identification patterns
- Patterns to enable backward and forward compatibility
  - Data Transfer Objects
  - Dataset Amendments
  - Tolerant Reader
  - Media Type Negotiation
  - Consumer-Driven Contracts
- Service governance concepts and practices
- What about SOA?

Web Service Infrastructure Patterns

- Service Interceptors and their uses
- Client Infrastructure patterns
  - Service Connector
  - Asynchronous Response Handler
  - Idempotent Retry
- What about SOA Infrastructures?
  - Web APIs and Service Registries, ESBs, and Orchestration Engines

Service Performance, Scalability, and Availability

- The CAP Theorem
- Asynchronous processing
- State management

See complete course description at **www.construx.com/learn**

Developer Boot Camp

On Site | On Demand | Blended Learning | Public

This intense hands-on seminar will give you the tools you need to be a professional software developer. This seminar combines techniques from the areas of design, construction, and testing to give you pragmatic guidance into the business of writing working code. Principles will be illustrated with numerous concrete examples of good and bad code in a variety of languages. In depth labs allow you to practice applying the principles.

Managing Complexity

- Essential and accidental difficulties
- Modularity
- Encapsulation
- Information hiding
- Coupling
- Cohesion
- Abstraction
- Interfaces
- Design by contract
- Knowledge as data

Designing for Change

- Variability analysis
- Typical changes
- Design to invariants
- Extension & intension
- Association versus inheritance
- Delay binding times
- Composability
- Open / closed principle

Key Construction Skills

- Design principles
- Design patterns
- Structured programming
- Object-oriented programming
- Functional programming
- Improving productivity
- Working on a team
- Writing legible code
- Tools, techniques, and practices

Increasing Your Software's Value

- Effective requirements
- Prototyping
- Modeling
- User interface design
- Read-time versus write-time convenience
- Characteristics of high value software
- Transparency

Error Handling

- Assertions
- Exceptions
- Diagnostics
- Defensive programming

Ensuring Correctness

- Functional unit testing
- Structural unit testing
- Automated feature testing
- Domain analysis
- Equivalence class partitioning
- Testing strategies
- Measuring complexity
- Minimize the lag between error insertion and error detection
- Avoiding common pitfalls
- XUnit frameworks
- Mutation testing
- Static analysis

See complete course description at **www.construx.com/learn**

See related course, Agile Developer Boot Camp for On-Demand Training

Code Complete Essentials

On Site · On Demand · Blended Learning · Public

In this intense one-day seminar you will learn dozens of proven tips, techniques, and principles to produce clean, industrial-strength code. Capturing the body of knowledge available from research, academia, and everyday commercial practice, this seminar synthesizes the most effective techniques and must-know principles into clear, pragmatic guidance. This seminar uses dozens of examples of good and bad code in Java, C++, C#, and Visual Basic to explain how to shorten development time, reduce errors, and make debugging easier. This seminar is taught by Steve McConnell, the coding guru who wrote the best-selling *Code Complete*, a computing industry classic that won the Jolt Excellence award for best programming book of the year and has been translated into more than a dozen languages.

Introduction
- Construction's critical role in software development
- Technology knowledge vs. principles knowledge
- Dealing with "accidental" and "essential" difficulties

Defensive Programming
- Error processing
- Effective strategies for anticipating change
- Code stepping
- Offensive programming

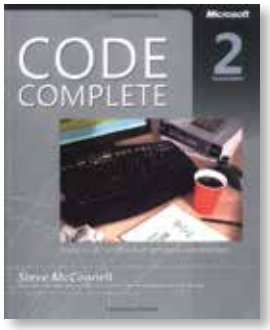
Creating High Quality Designs
- Differences in design effectiveness
- Attributes of great designers
- The Primary Technical Imperative: Managing Complexity
- Managing technical debt
- The relationship between naming and design
- Design guidance: information hiding, abstraction, encapsulation, modularization, cohesion, separation of concerns

High-Quality Routines
- Coding Horror: examples of low-quality routines
- Program layout techniques
- Low-effort, high-payoff commenting techniques
- The Pseudocode Programming Process

High-Quality Classes
- Good and bad reasons to create classes
- Designing interfaces

Code Optimization
- A defensive strategy for code optimization
- Three optimization approaches that don't work
- Example of intensive optimization

Quality Practices
- Debugging by superstition
- A scientific approach to debugging
- Tips for finding defects
- Tips for fixing defects
- Defect cost dynamics
- Error prone modules



See complete course description at **www.construx.com/learn**

Design Boot Camp

On Site · On Demand · Blended Learning · Public

Different designers will create designs that differ by at least a factor of 10 in the code volume produced. How do you invent simple, straightforward designs and avoid complex, error-prone ones? Understand the fundamental design principles that lead to high-quality designs requiring low implementation effort. Learn both Agile and traditional approaches to creating great designs quickly and economically.

What is "Design"?
- Design as an activity vs. a product
- Design as a tool for communicating
- Managing complexity with design
- Characteristics of excellent designs

Fundamental Design Principles
- Use abstraction
- Encapsulate design decisions
- Maximize cohesion; minimize coupling
- Design to invariants
- Avoid premature optimization
- Beware of Fisher's Fundamental Theorem

Managing Design Complexity
- Dimensions of design (interface, data, function)
- Measures of design complexity in each dimension
- Balancing local and global complexity

Design attributes: The "-ilities"
- ISO/IEC 9126 Quality Model
- Non-ISO/IEC 9126 "-ilities"
- Managing conflicts among the "-ilities"

Architectural Design
- Pipes & filters
- Model-view-controller
- Layered architectures
- Service-oriented architectures
- Blackboard

Design Paradigms
- Aspect-oriented design
- Object-oriented design
- Structured design
- Design patterns

Detailed Design
- Refactoring
- Semantics-preserving transformations
- Program design languages
- Error processing

Design Documentation
- Desirable properties of design documentation
- IEEE STD 1016-1998 Recommended Practice for Software Design Descriptions
- Agile design documentation

Creativity in Design
- Factors that contribute to excellent designs
- Value of creativity
- Research on software creativity
- Fudd's Law

See complete course description at **www.construx.com/learn**

"Arguably the best course I have ever taken." *Joe Johnson, Infospace*

Model-Based Design

On Site | On Demand | Blended Learning | Public

This two-day seminar provides in-depth, hands-on coverage of developing and documenting software designs from model-based functional requirements as part of a Model-Driven Architecture (MDA) software development approach. The software design is documented using a subset of the Unified Modeling Language (UML). Model-based designs are more complete, concise, less ambiguous than typical designs and lead to developing higher quality software with less cost and shorter schedules. No prior knowledge of object-oriented development, UML, or MDA is required. The companion seminar, Object-Oriented Analysis, explains how to start the MDA process and create the model-based requirements that feed into this design.

Introduction
- Why model?
- Software methods and models
- Introduction to UML
- Analysis vs. Design
- How to read an OOA

Fundamental Principles
- Importance of complexity
- Abstraction
- Encapsulation
- Design to invariants

Object Orientation
- Concepts and terminology of object orientation

Defining Interfaces
- Storyboarding
- Dialog maps
- Usability
- Interface control doucments (ICDs)

Deriving High-level Designs
- Deriving OOD classes and operations for OOA
- Deriving OOD classes and operations from Interface Designs

Low-level Design
- Deriving data structures from OOA
- Deriving procedure structures from OOA

Design Optimization
- Straight-forward optimizations
- High-level optimizations
- Low-level optimizations

Summary

See complete course description at **www.construx.com/learn**

Design Pattern Essentials

On Site | On Demand | Blended Learning | Public

Design patterns made understandable! Design patterns are powerful, predefined solutions to common software design problems. Patterns provide a powerful form of reuse because they are transportable across different languages and software architectures. This seminar introduces and explains the highest-leverage design patterns in a clear way that's easy to understand. You will learn how to use specific design patterns to improve your designs, and you will practice applying design patterns to sample design problems. You will also see how design patterns can improve your design process overall.

Introduction
- Design patterns defined
- Understanding design pattern forces

Fundamental Design Principles
- Maximize cohesion, minimize coupling
- Encapsulate design decisions
- Base the design on the invariants
- Subclasses should be substitutable
- Favor association over inheritance
- Minimize overall complexity

UML Notations
- Class, attribute, operation
- Association and multiplicity
- Aggregation and composition
- Inheritance
- Abstract class

High Leverage Patterns
- Model-View-Controller
- Adapter
- Façade
- Bridge / Strategy
- Composite
- Observer
- Template method
- Factory method
- Abstract factory
- Singleton

Other Useful Patterns (as time permits)
- Iterator
- Proxy
- Decorator
- Command
- State
- Data access object

See complete course description at **www.construx.com/learn**

> **"The instructor was very engaging! He kept my interest the whole time and is the perfect combination of expert, geek, and comedian."**
>
> *Heidi Shea, Newmarket International*

Professional Tester Boot Camp

On Site | COMING IN 2017 | COMING IN 2017 | Public

How do professional testers test software? This seminar teaches the techniques, tips, tricks, and strategies used by test professionals. You will learn and apply numerous detailed test design strategies for black-box (functional) testing and system testing. You will also learn practical techniques for planning, designing, and executing effective testing on real software projects. Determine how much testing is enough for your project, whether your test cases are adequate, and how to minimize wasted testing effort. This seminar emphasizes black box (functional) testing and system testing.

Test Planning
- How to make your testing more effective
- Making your testing more efficient
- Evaluating test case designs
- When to plan for testing
- Relating testing to other software development activities
- Evaluating test case designs
- Determining how much testing is enough

Domain Testing
- Where to find domain specifications
- Analyzing domain specifications
- Common domain defects you should test for
- Designing input domain tests
- Designing output domain tests
- Analyzing domain boundaries

Testing from Requirements
- How to ensure that every requirement is tested
- Deciding the number of test cases necessary for each requirement
- Creating test cases from use cases
- Creating test cases from relationship models
- Creating test cases from state-transition models

Tool Support for Testing
- Determining which tools are useful for supporting testing activities

Pairwise Testing
- When is pairwise testing useful?
- Learn how to reduce the number of test cases when you have to test all combinations.
  For example:
  - Multiple operating systems
  - Multiple web browsers
  - Plug-ins
  - Web server software
  - Server operating systems
- Downloadable software to figure out your test cases
- Commercial tools that can help with pairwise testing

Test Automation
- Improving the efficiency of testing with automation
- Why is test automation harder than just recording and playing back keystrokes?
- How to get started with data driven test automation
- Advantages of keyword driven test automation
- Commercial tools/frameworks to support keyword driven test automation

Testing the Tests
- How do you know your tests are reliable?
- How do you know your tests are effective?
- How do you uncover and improve low quality tests?

See complete course description at
www.construx.com/learn

Agile Testing for Software Developers

On Site | On Demand | Blended Learning | Public

Test Driven Development is a critical component of Agile development. Unit testing without TDD typically achieves only 50-60% statement coverage, but with TDD you will achieve 100% branch coverage. Learn how to design efficient and effective tests and how to make the software itself more testable. See how to plan and execute an effective developer testing strategy. Understand the various ways of measuring the coverage and effectiveness of your test cases.

What is the Test Driven Approach?

Unit Testing Fundamentals

Unit Testing Patterns and Anti-Patterns
- Stubs, drivers, shunts, mocks
- Data-driven, delta, exceptions, guards

Domain Analysis
- Equivalence classes
- Boundary values
- Pairwise testing

Structural Analysis
- Control flow testing
- Coverage metrics
- Complexity metrics

High Level TDD
- Agile Acceptance Testing
- Testing an MVC pattern

Writing Testable Code

Testing with Contracts

Expository Programming
- Variable names
- Routine names
- Intentional programming
- Comments

Modularity
- Encapsulation
- Information hiding
- Interfaces
- Abstraction
- Coupling
- Cohesion

Design Patterns

Functional Design

Legacy Code
- Enhancing
- Fixing
- Refactoring

See complete course description at
www.construx.com/learn

> **"Excellent seminar and great presenter. He made sure we understood the topic before moving on."**
>
> *Dian Von Ballmoos, Fandango*

Developer Testing Boot Camp

On Site · On Demand · Blended Learning · Public

Developer testing is a critical component of software development--yet studies show that developer testing is typically inefficient and ineffective. Significant developer effort is spent without finding defects that should have been found. Learn how to be more efficient and effective by achieving higher test coverage and how to make the software itself more testable. See how to plan and carry out an efficient and effective developer testing strategy. Avoid common testing pitfalls, and learn to determine how much developer testing is enough. 2-day seminar.

Core Concepts
- Test case, test set, test suite
- Positive and negative test cases
- Functional and structural testing
- Unit, integration, system, acceptance, regression testing
- Test coverage
- Test tools and frameworks
- Testing as risk mitigation

Functional Testing
- Testing from requirements
- Requirements coverage
- Input domain analysis
- Output domain analysis
- Equivalence class partitioning
- Equivalence class coverage
- Boundary value analysis
- Boundary value coverage
- Combinatorial analysis
- All pairs coverage

Structural Testing
- Control flow analysis
- Statement coverage
- Decision (branch) coverage
- Condition-decision coverage
- Multiple condition coverage
- Modified condition/decision coverage
- Path coverage
- Loop coverage
- Static Analysis
- Cyclomatic complexity
- Fan in
- Fan out

Test Planning
- Determining the ROI of testing
- Cost/benefit of automating tests
- Cost/benefit of unit testing
- Cost/benefit of integration testing
- Cost/benefit of system testing
- Cost/benefit of designing for testability
- Introduction to risk-based testing

Unit testing tools and practices
- Unit testing frameworks
- Stubs and drivers
- Test doubles and mock objects
- Data driven tests

Test driven development
- Process of test driven development
- Benefits of test driven development
- Costs of test driven development

See complete course description at **www.construx.com/learn**

Total Project Quality

On Site · On Demand · Blended Learning · Public

From project inception, a focus on quality through planning, execution, and delivery can help improve software project cost, schedule, and functionality. This three-day seminar shows you how to define quality in specific terms with a focus on how to decide what is "good enough" for a particular project. You will learn to align project activities to achieve quality throughout the project lifecycle—including numerous alternatives to end-of-project testing. [17 PDUs]

Defining Quality
- Textbook quality definitions
- QA vs. QC

Quality Goals
- The ISO 9126 definition of quality
- Characterization
  - Scaling a characteristic
  - Prioritization of characteristics
- Establishing what it means to be "good enough"

General Strategy
- The basic philosophy of software quality
- Early removal of defects
- Insulating from impact of defects
- Using the PDCA cycle

Finding Faults
- Common faults in software and where they are created
- Identifying the defects
- Tracking defects

Detection Toolbox
- Peer reviews
- Quality attribute workshops
- Dynamic testing
- Prototyping
- Detection effectiveness

Approaching Quality
- Basic approach
- Lifecycles and characteristics

Implementing Quality in Requirements
- Five whys
- Task analysis
- Vision & goals
- Gists
- Fit criteria

Implementing Quality in Design
- Facilitated workshops
- Architecture tradeoff analysis method
- Design impact estimation
- Personas

Implementing Quality in Code
- Satisfice and softgoals
- Optimization

Analytical Toolbox
- Measures
- Comparing discrete classes
- Trending a class
- Process control
- Collecting data

Managing Software Quality
- Organizational level quality
- Process improvement

Establishing a Quality Culture
- Growing capability
- Selecting practices
- Audits

See complete course description at **www.construx.com/learn**

10X Software Engineering - Teams

On Site · On Demand · Blended Learning · Public

Decades of research have found at least a ten-fold (10X) difference in productivity and quality between the best developers and the worst—and between the best teams and the worst. Discover the 8 Key Principles of 10X Engineering. Gain a deeper understanding of the factors that affect productivity and avoid the productivity traps of "minus-x" engineering. Learn and apply critical techniques that will turn your team into a high performing, 10X Team. [18 PDUs]

Defining 10X
- 10x differences in productivity and quality
- 10x principles

10X Principle: Avoid Minus-X Engineering
- Classic mistakes
- Brute force quality
- Multi-tasking
- Typical Minus-X project
- Your Minus-X project

Basic Engineering
- Mastery of fundamentals and excellent execution
- Technical fundamentals
- Technical management fundamentals
- Quality fundamentals
- Your organization's bare essentials

10X Principle: Seek Ground Truth
- Daily build & smoke test
- Project tracking
- Communicating status
- Root cause analysis
- Change control–Agile and plan driven
- Gates and checkpoints
- Evidence-based case study

10X Principle: Base Decisions on Data
- Plan-Do-Check-Act
- Measurement
- Iteration and incrementalism
- How much agility is enough?
- Customer involvement
- Power of interactive workshops

10X Principle: Tailor the Solution to the Problem
- Streamlining work with intellectual phase profiles
- Phase and activity variations
- Life cycle models
- Efficient information capture
- Documents and digital cameras
- Toolboxes

10X Principle: Set Direction

Project Direction
- Project charter
- Vision statements

10X Principle: Set Direction

Feature Selection
- Pareto analysis
- T-shirt sizing
- Rolling wave planning
- Product backlog

10X Principle: Attack Uncertainty
- Responses to uncertainty
- Risk management
- Accurate estimates
- Uncertainty and iteration

10X Principle: Minimize Unintentional Rework
- Why would any rework be intentional?
- Early defect detection
- Defect removal rates for specific techniques
- Comparison of iterative and sequential defect removal effectiveness
- Formal inspections

See complete course description at **www.construx.com/learn**

10X Individuals and Interactions: Unleashing the Power of the Team

On Site · On Demand · Blended Learning · Public

The Agile Manifesto states that, "We value individuals and interactions over processes and tools." What does that really mean? This workshop explains what that means by diving into the details of human interactions, including working with different personalities, dealing with conflict, collaborating effectively, team decision making, creating shared vision, and building high performance teams. It is complementary to the more process-focused content in Agile practices like Scrum, Kanban, or Construx's 10x Software Development programs.

Software projects rarely fail for technical reasons. They fail for reasons related to *people*. Make sure the people on your next project have the human interaction skills they need to succeed!

High Performance 10X Teams
- Team basics
- Characteristics of high performance teams
- Role of teamwork
- Effective vs. ineffective teams
- Stages of team development
- Guidelines for team members and leaders
- Self-Assess your teamwork skills
- Action Plan

Leveraging Diverse Personalities to Form High Performance Teams
- Personality types
- Social Style test
- Interpretation of Social Style test results
- Drivers (Lions)
- Expressives (Otters)
- Amiables (Golden Retrievers)
- Analyticals (Beavers)
- Personality substyles
- Communicating with different social styles
- Conflict resolution with different social styles
- Working in teams with different social styles
- Action Plan

Supporting Interactions with the RULER Model
- Three steps to individual self-regulation
- Working with self-regulation in others
- Supporting high performance teams with emotional intelligence
- Gaining deeper insight into yourself with the Mood Meter
- Using the Mood Meter to support productive meetings
- Creating team charters with the RULER model
- Identifying desired steady state with the RULER charter
- Identifying necessary transitory states with the RULER charter
- Action Plan

High Performance Communication
- Elements of high performance communication
- Preparing for difficult conversations
- Five step guide to addressing conflicts
- Crucial Conversations
- Making a plan for a Crucial Conversation
- Succeeding with presentation to executives
- Preparing for difficult conversations using Social Styles
- Action Plan

High Performance Collaboration

High Performance Team Decision Making

Maximizing Energy and Personal Effectiveness

See complete course description at **www.construx.com/learn**

Our Advantage Is Our Experts

Steve McConnell is CEO and Chief Software Engineer at Construx Software where he consults to a broad range of industries, teaches seminars, and oversees Construx's software development practices. Steve is best-known as the author of *Code Complete*. He has also written *Software Estimation, Professional Software Development,* and other titles. His books twice won the Jolt Excellence award for outstanding software development book of the year, as well as Amazon.com's award for best computer book of the year.

Steve personally oversees all course development and delivery at Construx, guaranteeing your team the best possible use of your valuable training time.

Earl Beede is a Senior Fellow at Construx Software, where he designs and leads seminars and provides consulting services on early project-lifecycle practices, estimation, requirements, quality assurance, contract management, and software methodologies. With more than 20 years experience in quality assurance, systems analysis, process architecture, and management, Earl has designed and written software development processes for a wide variety of industries. He is a member of the IEEE Computer Society and a coordinator of the Seattle Area Software Process Improvement Network.

John Clifford is a Senior Fellow at Construx Software where he focuses on software development, project management, and team management practices with an emphasis on Agile practices. With more than three decades of IT experience, John has held leadership roles as a development engineer, product feature team manager, group QA manager, group project manager and development director. John holds Certified Scrum Master, Certified Scrum Product Owner, and Certified Scrum Practitioner certifications from the Scrum Alliance.

Melvin Perez-Cedano is a Senior Fellow at Construx Software where he focuses on software design, software construction, software process, and configuration management. He has a successful track record transforming dysfunctional and immature organizations into performance and quality-oriented environments. Melvin has deep experience in Agile practices including code cleaning, refactoring, and unit testing, and is also well versed in UML, RUP, and CMMI. He has lectured extensively throughout the Americas and is a former IEEE Distinguished Lecturer for Latin America.

Jenny Stuart is the Vice President of Consulting at Construx Software. Jenny applies her 20+ years of leadership experience in software testing, process improvement, quality assurance, and Agile techniques to helping companies make dramatic improvements in their software development processes. She holds a BSEE from the University of Washington, and CSDP certification. Jenny is a member of the Computer Society of the IEEE, the Association for Computing Machinery, the American Society for Quality, and the Quality Assurance Institute.

Steve Tockey is the author of *Return On Software*, a guide for companies that want to maximize their investment in software development. As a Principal Consultant at Construx, Steve focuses on software project management, estimation, software quality, object-oriented development, and distributed object computing. He is a Certified Software Development Professional (CSDP), and chairs the CSDP Certification Committee of the IEEE Computer Society. He is also a charter member of the OMG, the group that oversees development of the UML.

Bob Webber is a Senior Fellow at Construx with over 30 years of software experience delivering innovative products in telecommunications, entertainment and life sciences. Bob's roles have ranged from software developer to executive R&D positions at GTE and AT&T, followed by engineering and product management leadership at three successful start-up companies. As CEO, Bob led TranSenda International to successful acquisition by BioClinica (BIOC), where he became Vice-President of Product Management.

Scrum.org™

Project Management Institute Registered Education Provider

IIBA® Endorsed Education Provider™

Construx®

www.construx.com