Software Architect Career Path

Level 1: Developer
Level 2: Senior Developer
Level 3: Technical Lead
Level 4: Associate Architect
Level 5: Software Architect

Steve McConnell, CEO, Construx Software

Jenny Stuart, Vice President Consulting, Construx Software

Version 4.0.51, October 2018

Construx

Copyright

Construx Software Builders, Inc.
10900 NE 8th Street, Suite 1350
Bellevue, WA 98004
U.S.A.

This paper is for informational purposes only. This document is provided "As Is" with no warranties whatsoever, including any warranty of merchantability, noninfringement, fitness for any particular purpose, or any warranty otherwise arising out of any proposal, specification, or sample. Construx Software disclaims all liability relating to use of information in this paper.

Contents

Introduction to Construx Career Pathing and the Professional Development Ladder

Unlike established professions, career paths for software professionals have never been well defined. Adopting and executing a well-defined software career path presents a significant opportunity. For individuals, career pathing supports enhancing their personal capabilities and advancing in their careers. For companies, career pathing allows for increasing staff capability and capacity over time without adding staff.

Construx believes that a software professional's experiences should accumulate and build in ways that tap into each person's ability to grow over time and that result in steadily increasing contributions to their organizations.

Construx's Professional Development Ladder (PDL), was first deployed in 1998. Since then this program has been deployed at companies worldwide and has been regularly updated.

Professional Development Ladder White Paper

This paper describes one specific career path using Construx's career pathing program. The full program professional development program is described in Construx's "Career Paths for Software Professionals" white paper. See the white paper at https://www.construx.com/resources/white-papers/.

Focus on Best Practices

After spending more than 20 years working with hundreds of leading companies worldwide, Construx has found that capability for software professionals is like a three-legged stool in which the legs are:

- Technology Knowledge
- Business Domain Knowledge
- Software Development Best Practices Knowledge

Technology Knowledge tends to be acquired by software professionals automatically, through on-the job, hands-on technology work, as does Business Domain Knowledge.

Software Development Best Practices Knowledge is the understanding of principles that apply regardless of which detailed technology is used and regardless of the specific business area. Research for decades has shown an overwhelming return on investment for acquiring capability in software development best practices. Development of capability in this area is the focus of Construx's Professional Development Ladder.

Why Choose Construx's PDL?

Version 1.0 of Construx's Professional Development Ladder was first deployed within Construx and released publicly more than 20 years ago. Since then, the ladder has been deployed at companies in the US and throughout the world. We have steadily incorporated experiences implementing the ladder into successive versions; the ladder has continued to evolve and improve to meet the needs of software professionals and their organizations.

Defining career paths for software professionals using Construx's PDL provides numerous professional development benefits for technical individuals and their organizations:

- Rapid development of technical staff capabilities—significantly faster than is possible without the structure and focus of the PDL.
- Structure and support for individual career paths that goes far beyond what individual software professionals are able to create on their own.
- Structure and detail that guides managers in their coaching of technical staff members.
- Career guidance based on industry standards using a well-thought-out, tested, highly mature development framework.
- Improved retention of technical staff, who see their organization investing in them via the PDL.

In addition to the obvious benefits, an organization that commits to using the PDL will realize significant additional benefits that are arise from the shared understanding of professional development:

- Improved ability to assess new staff members and onboard them quickly and effectively, due to the shared understanding of software skills that is provided by the PDL.
- Ability to maintain a technical skills inventory, using the structure of the PDL.
- Ability to perform staff capacity planning, using the structure of the PDL.
- Ability to set clear expectations about advancement and promotion, based on shared, specific, structured understanding of the professional development activities needed to support advancement (if the organization chooses to use the PDL that way).

The PDL's Building Blocks for Software Career Pathing

To provide career pathing for software professionals, Construx's PDL is comprised of several building blocks:

- Software development Knowledge Areas (KAs), based on software industry standards
- Capability Levels
- A Professional Development Matrix (PDM) created by combining KAs and Capability Levels
- Professional development activities needed to attain each combination of KA and Capability Level
- Role-specific career paths, built using the KAs, Capability Levels, and professional development activities described above.

Software Development Knowledge Areas

Construx's software development Knowledge Areas (KAs) define the body of knowledge that software professionals should understand and be able to apply. The KAs are based on the Software Engineering Body of Knowledge (SWEBOK) knowledge areas and are enumerated below.

- Configuration Management (CNFG)
- Construction (CNST)
- Design (DSGN)
- Foundations (FNDN)
- Maintenance (MNTC)
- Models and Methods (MTHD)
- Process (PRCS)
- Management (MGMT)
- Quality (QUAL)
- Requirements (RQMT)
- Testing (TEST)

Capability Levels for Software Professionals

Construx recognizes three levels of capability in career development: Introductory, Competence, and Leadership. Each of these levels includes defined performance standards:

Introductory The professional performs or is capable of performing basic work in an area, generally under supervision. The professional is taking effective steps to develop his or her knowledge and skills.

Competence The professional performs effective, independent work in an area, serves as a role model for less expert professionals, and occasionally coaches others.

Leadership The professional performs exemplary work in an area. The professional regularly coaches professionals and provides project-level leadership and possibly company-wide leadership. The professional is recognized within the company as a major resource in a defined area.

The Centerpiece of the PDL: The Professional Development Matrix

The centerpiece of Construx's Professional Development Ladder is an 11x3 Professional Development Matrix (PDM) that is produced when the KAs and Capability Levels are combined. This is illustrated graphically in Figure 1.

Figure 1 The Professional Development Matrix (PDM)

	Knowledge Area										
Capability Level	Configuration Management	Construction	Design	Foundations	Maintenance	Models and Methods	Process	Management	Quality	Requirements	Testing
Introductory											
Competence											
Leadership											

The matrix arising from the combination of the 11 standards-based Knowledge Areas and the 3 defined Capability Levels provides an organization for career development that is simultaneously highly structured and highly flexible.

Professional Development Activities

In Construx's PDL, each "cell" in the Professional Development Matrix consists of the activities needed to achieve performance at that level of capability. We believe that true career development for software professionals must consist of the combination of notional learning (books), hands-on learning (classroom or self-paced training) and real-world experience. Thus our PDL contains these types of activities:

- Reading
- Self-paced training or classroom training
- Defined experience

We have found that being exposed to similar learnings on each topic in reading, training, and experience provides important reinforcement.

Reading, and Types of Reading

Reading is a key part of career development, and we believe that true professional development cannot occur without focused reading beyond short articles on the internet. In our ladder work, we differentiate between three types of reading: "Inspectional," "Analytic," and "Syntopical."[1]

Inspectional reading is used to attain a quick overview of an article or book. *Analytic* reading is what we normally think of when we think of reading, i.e., deep reading with intent to fully understand material. *Syntopical* reading refers to reading a collection of related materials together, for the purpose of comparing, contrasting, and extracting more meaning from the material than could be acquired by reading any one source on its own.

Training

Training is also a key part of career development. Online self-paced training now allows individuals to obtain professional development at times and places that are most convenient for them, without the delays associated with classroom training. The detailed activities in this document focus on self-paced training available from Construx OnDemand. Equivalent classroom training may be substituted when available.

Defined Experience

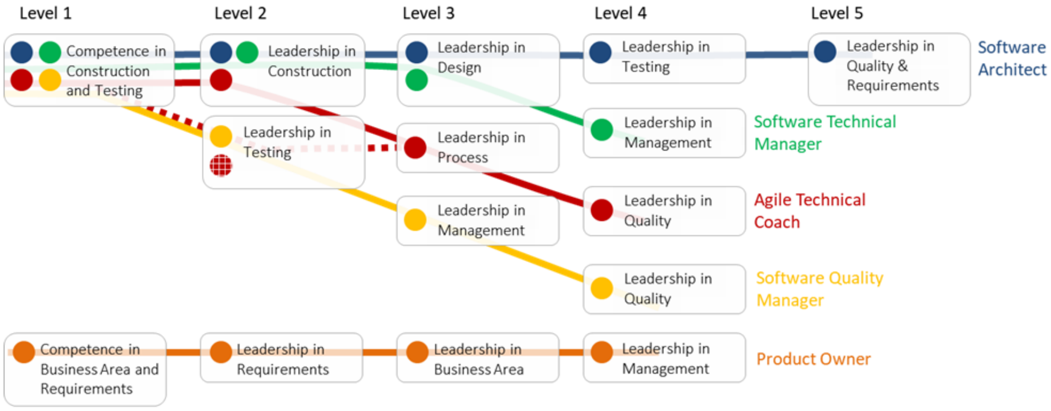
Development of truly enhanced capability in a software professional cannot be accomplished without extensive hands-on experience. Experience must be obtained deliberately, in ways that add up over time, which is why Construx's PDL requires *Defined Experience*.

[1] See "How to Read a Technical Article," Steve McConnell, *IEEE Software*, Nov/Dec 1998. (link)

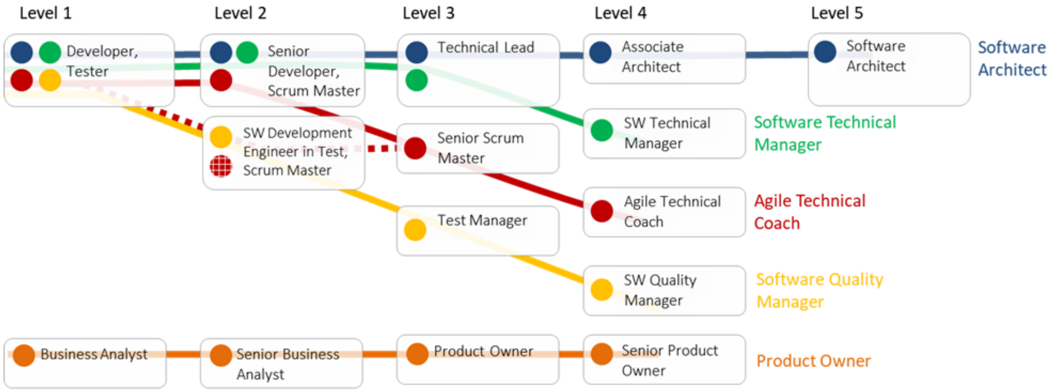
Path Decision Points

Construx's software career paths are designed to support structure, flexibility, and options during a software professional's career. To the maximum degree possible, required reading, training, and activities are designed to overlap at lower levels of each path on the ladder and become more specific at higher ladder levels. This allows a professional to explore multiple career options before fully committing to one specific path.

As the diagram below shows, the work for Software Architect, Software Technical Manager, Agile Technical Coach, and Software Quality Manager is very similar at Level 1 on their respective paths. The Product Owner path is the only path that is significantly different at Level 1.



For additional context, the following diagram shows the path options with representative titles that would be achieved upon completing each level (with the caveat that specific titles vary considerably from one organization to another):



After Level 1, the top paths diverge into a development focus and a testing focus. With the development-focused path at Level 2 a software professional can continue the possibility of eventually pursuing the Software Architect, Software Technical Manager, or Agile Technical Coach path by doing the same work.

With the testing-focused path at Level 2, a software professional can continue the possibility of eventually pursuing the Software Quality Manager or Agile Technical Coach by doing the same work.

At Level 3, the Agile Technical Coach path diverges (meaning the Level 3 work is different from other paths). On the top paths, the same Level 3 work can still support an ultimate goal of either Software Architect or Software Technical Manager.

The fact that the paths diverge does not mean it is impossible for a professional to change course later. It just means that more backfilling (via a combination of reading, training, and experience) will be required when that occurs.

The combination of structure and flexibility in Construx's career paths means that professionals who develop themselves using the PDL will realize benefit from the investment they have made in themselves even if they change course later in their careers.

Software Architect Career Path Structure and Overview

Using the structure provided by the Professional Development Matrix, a professional software developer will progress through a series of five Career Path levels on the Software Architect career path. The ultimate goal of a software architect is the capabilities shown in Figure 2.

Figure 2 Full requirements for the Software Architect path

	Knowledge Area										
Capability Level	Configuration Management	Construction	Design	Foundations	Maintenance	Management	Models and Methods	Process	Quality	Requirements	Testing
Introductory	●	●	●	●	●	●	●	●	●	●	●
Competence	●	●	●	●	●		●		●	●	●
Leadership		●	●						●	●	●

The general pattern of progressing through the Career Path Levels looks like this:

Level 1 (Developer) Acquire Introductory capability in all KAs and Competence in 2 KAs. Expected time in level is 1-2 years (nominally 2).

Level 2 (Senior Developer) Acquire Competence in 2 additional KAs and Leadership in 1 KA. Expected time in level is 1-2 years (nominally 2).

Level 3 (Technical Lead) Acquire Competence in 2 more KAs and Leadership in 1 more KA. Expected time in level is 1-2 years (nominally 2).

Level 4 (Associate Architect) Acquire Competence in 2 more KAs and Leadership in 1 more KA. Expected time in level is 1-2 years (nominally 2).

Level 5 (Software Architect) Acquire Competence in 1 more KA and Leadership in 2 more KAs. Expected time in level is 1-2 years (nominally 2).

The expected time to complete all five levels in this career path is 8-10 years. Not every professional will choose to advance through all five levels. Many will choose to plateau at Level 3 or Level 4; indeed most organizations do not need every technical staff member to advance beyond Level 3.

Development of capability in each cell depends on acquisition of both relevant *Knowledge* and *Experience*. We have found that if either Knowledge or Experience gets too far ahead of the other overall professional development suffers. Construx's career paths keep acquisition of Knowledge and Experience in balance throughout each of our career paths to support effective and rapid development of professional capabilities.

In many organizations, "Architects" have 20+ years of experience, and so some people might be surprised that a professional will reach the Technical Lead level after only 5-6 years, Associate Architect after 6-8 years, or Software Architect after 8-10 years. But this is the power of focused professional development. If left to chance, a person can work 20+ years and still not have the level of capability that focused professional development will provide after only 8-10 years.

Common Variations

The path described in this document focuses on the knowledge areas of Construction, Design, and Testing and ultimately adds in Quality and Requirements. In other words, this path begins with a Developer orientation and moves over time toward Architecture

Common alternatives to the nominal path include:

- A path that puts more emphasis on Models and Methods and less on Requirements
- A path that puts more emphasis on the Business Area
- A path that puts more emphasis on Management and less on Testing

These variations are summarized in the "Customization and Adaptation" section at the end of this document

The remaining sections in this white paper describe the nominal Career Path in more detail.

Level 1: "Developer"

While working on Level 1 the professional will complete Introductory work in all KAs and achieve Competence in Construction and Testing. The focus on completing Introductory work in all KAs at Level 1 is based on the traditional approach to engineering education in which breadth is obtained prior to depth. Depth will be acquired incrementally in higher levels. The focus on Competence work in Construction and Testing lays the foundation for eventual work toward Leadership in these two areas. At this level, the requirements for the Software Architect path, Agile Technical Coach path, and Software Technical Manager path are very similar, which provides the option for a professional to complete the Level 1 requirements on this path and still switch to a Software Technical Manager or Agile Technical Coach path at a higher level. The expected time to complete this level is 1-2 years.

Figure 3 Requirements to complete Level 1 on the Software Architect Path

	Knowledge Area										
Capability Level	Configuration Management	Construction	Design	Foundations	Maintenance	Management	Models and Methods	Process	Quality	Requirements	Testing
Introductory	●	●	●	●	●	●	●	●	●	●	●
Competence		●									●
Leadership											

Level Proficiency Goals

Upon completion of this level, this person will have numerous proficiencies including skill with at least two programming languages / technologies; code structure, naming, and commenting techniques; data structure implementation techniques; database implementation techniques; interface/API implementation techniques; advanced debugging techniques; automated feature test techniques; boundary and equivalency test case techniques; output and breakpoint debugging techniques; regression testing, single step, unit, and integration test techniques; test design techniques, and unit, component, and integration test techniques.

The following sections describe the detailed activity requirements for completing this level.

Level Terminology

In this document, “working in a level” refers to the work needed to complete a level. A “Level 2 professional” (someone who has completed the work for Level 2) will be “working on Level 3.” The professional will become a “Level 3 professional” upon completing Level 3 work.

Reading

Introductory

The following reading provides basic knowledge in all knowledge areas and sets the stage for more advanced reading in selected areas.

The order of reading will vary from one person to another and should be defined in each individual’s Professional Development Plan (PDP). For more on the PDP, see the section at the end of this document.

- "A brief history of project management (and why you should care)", Chapter 1 in Making Things Happen, Scott Berkun (20 pages, Analytical) (MGMT)
- "Software Engineering Code of Ethics and Professionalism", ACM/IEEE-CS (3 pages, Analytical) (FNDN)
- "What Is Software Testing? And Why Is It So Hard?", James A Whittaker (10 pages, Analytical) (TEST)
- Code Complete 2nd Edition, Chapter 20, Steve McConnell (25 pages, Analytical) (QUAL)
- Code Complete 2nd Edition, Chapter 24, Steve McConnell (25 pages, Analytical) (MNTC)
- Code Complete 2nd Edition, Chapter 5, Steve McConnell (45 pages, Analytical) (DSGN)
- Code Complete 2nd Edition, Chapters 1, 2, 8, 34, Steve McConnell (66 pages, Analytical) (CNST)
- Software Requirements 3rd Edition, Chapter 1, Karl Wiegers (22 pages, Analytical) (RQMT)

Competence

The following reading supports Competence knowledge in the areas of Construction and Testing.

- A Friendly Introduction to Software Testing, Bill Laboon (230 pages, Analytical) (TEST)
- Code Complete 2nd Edition, Steve McConnell (960 pages, Analytical) (CNST, DSGN, MNTC)
- Technology specific unit testing book such as the Art of Unit Testing with Examples in C#, Roy Osherove; Pragmatic Unit Testing in Java8 with Junit (200-300 pages, Inspectional) (TEST)

Training

Training Support for Construx Career Paths

The training described in this career path can be completed via the online, self-paced training available through Construx OnDemand. For more information see https://ondemand.construx.com.

Introductory

The following training supports development of Introductory capability in all knowledge areas and sets the stage for more advanced capability in selected areas.

As with reading, the order of training will vary from one person to another and should be defined in each individual's PDP.

- Agile Planning and Estimation (~2 hours) (MGMT)
- Code Complete, Sections: Design, Design Tools, Complexity, Design Tools For Complexity (~2 hours) (DSGN)
- Code Complete, Sections: Introduction, Defensive Programming, Complexity (~1 hours) (CNST)
- Introduction to the SWEBOK (~1 hour) (FNDN)
- Kanban Overview (~3 hours) (PRCS, MGMT)
- Rational Design Process: How and Why to Fake It, Construx Braincast (~1 hour) (DSGN)
- Scrum Overview (~3 hours) (PRCS)
- Understanding Software Project Lecture Series - "Core" Lectures (~2 hours) (QUAL, FNDN)
- Understanding Software Project Lecture Series - "Lifecycle" Lectures (~3 hours) (MTHD, FNDN, PRCS)

Competence

The following training supports Competence capability in the areas of Construction and Testing.

- Agile Practices for Developers (~2 days) (CNST)
- Code Complete Essentials (~1 day) (CNST, MNTC)
- Developer Testing Boot Camp (~1.5 days) (TEST)
- One or more approved technology courses from Pluralsight or LinkedIn Learning (or equivalent) (~1 day) (CNST)

Work Activities

Introductory

The following experiences provide important context that support development of the full capabilities of a software professional. The list of activities is lengthy, but most activities are introductory in nature and can be completed fairly quickly.

As with reading and training, the order of work activities will vary from one person to another and should be defined in each individual's PDP.

Configuration Management-Focused

- Learn a version control tool (CNFG)
- Participate in the release process for a significant project (CNFG)
- Review a project's build environment (CNFG)
- Review and discuss project artifacts related to Configuration Management (CNFG)

Construction-Focused

- Learn a programming language / technology (CNST)
- Learn multiple utilities or tools (CNST)
- Participate in a code review (CNST)
- Review a sampling of code that implements critical functionality for your product or organization (target of ~1000 lines of code, ~20 pages) (CNST)
- Review and discuss project artifacts related to Construction (CNST)

Design-Focused

- Learn a design technique (DSGN)
- Participate in a design process and artifact creation (DSGN)
- Participate in a design review (DSGN)
- Review and discuss project artifacts related to Design (DSGN)

Foundations-Focused

- Review a break-even analysis or optimization analysis (FNDN)
- Review a project's business case or similar economic analysis (FNDN)
- Review an economic analysis involving allowance for inaccuracy, risk, or uncertainty (FNDN)
- Review current work practices and evaluate how well they comply with the "Software Engineering Code of Ethics and Professionalism" (FNDN)

Maintenance-Focused

- Participate in a fix in a production system (MNTC)
- Review and discuss project artifacts related to Maintenance (MNTC)
- Review results of debugging an issue in a production system (MNTC)

Management-Focused

- Learn a structured group estimation technique (e.g. planning poker, wide-band Delphi, etc.) (MGMT)
- Learn an experience-based, quantitative estimation technique (MGMT)
- Participate as a reviewer in a formal review of one or more project artifacts in Management (MGMT)
- Review and discuss project artifacts related to Management (MGMT)

Models and Methods-Focused

- Review and discuss project artifacts related to Methods (MTHD)
- Review the way in which design is represented on a project, focusing on modeling approach and notation (MTHD)
- Review the way in which requirements are represented on a project, focusing on modeling approach and notation (MTHD)

Process-Focused

- Learn Scrum, Kanban, or other Agile framework, process, or approach (PRCS)
- Participate as a reviewer in a formal review of one or more project artifacts in Process (PRCS)
- Participate in a daily Scrum / daily Kanban (PRCS, MGMT)
- Participate in a retrospective (PRCS, MGMT, QUAL)
- Participate in a sprint review (or equivalent) (PRCS, MGMT)
- Participate in a story point estimation session (planning poker, affinity, etc.) (or equivalent) (PRCS, MGMT)
- Participate in sprint planning (or equivalent) (PRCS, MGMT)
- Review and discuss project artifacts related to Process (PRCS)

Quality-Focused

- Learn your organization's defect classification definitions/technique (QUAL)
- Review a project's defect classification approach (QUAL)
- Review a project's quality metrics (QUAL)
- Review a project's quality-related release criteria (QUAL)

- Review a quality plan (QUAL)
- Review a team's Definition of Done (or equivalent) (QUAL)
- Review and discuss project artifacts related to Quality (QUAL)

Requirements-Focused

- Learn a requirements specification technique (user stories, Planguage, use cases, etc.) (RQMT)
- Participate in a refining meeting (or equivalent) (RQMT)
- Participate in a requirements discovery session (interview, job shadowing, workshops, etc.) (RQMT)
- Review a project's user stories, use cases, or other requirements (RQMT)
- Review and discuss project artifacts related to Requirements (RQMT)

Testing-Focused

- Create a defect report for three different types of failures (TEST)
- Learn a test technique (TEST)
- Participate in a regression testing effort (TEST)
- Participate in testing during a sprint/increment (TEST)
- Review a set of test cases (TEST)
- Review and discuss project artifacts related to Testing (TEST)

Competence

The following experiences ensure development of both broad and deep capability in the areas of Construction and Testing.

Construction-Focused

- Create 2 or more utilities or tools (CNST)
- Create multiple APIs (CNST)
- Create multiple production modules (CNST)
- Create multiple user interfaces (CNST)
- Create one or more database designs (CNST)
- Discuss approaches to debugging with multiple senior developers and architects (CNST)
- Lead a review and discussion of project artifacts related to Construction (CNST)
- Participate as a reviewer in a formal review of one or more project artifacts in Construction (CNST)

- Participate in multiple collaborative construction efforts (pair programming, peer review, etc.) (CNST)
- Participate in performance profiling / tuning (CNST)

Testing-Focused

- Create and execute test cases for multiple user stories (or equivalent) (TEST)
- Create automated functional tests for multiple user stories (or equivalent) (TEST)
- Create automated unit/integration tests for multiple modules (TEST)
- Create test cases using boundary value analysis (TEST)
- Create test cases using equivalence class analysis (TEST)
- Create/setup a test environment (TEST)
- Participate as a reviewer in a formal review of one or more project artifacts in Testing (TEST)
- Participate in creation of a test strategy / test plan (TEST)
- Participate in smoke testing or adding automation to a BVT/smoke test (TEST)
- Perform regression against a defect fix (TEST)

Level 2: “Senior Developer”

By completing Level 2 the professional will achieve Competence in Configuration Management and Design and Leadership capability in Construction. Although this path diverges slightly from the Software Technical Manager path and the Agile Technical Coach path at this level, this focus still leaves open the possibility of shifting to either of those paths if desired, while still laying the groundwork for eventually obtaining leadership in the Design KA.

At this level, the professional has a fairly strong background in software engineering and can work independently as needed. This person is expected to be a strong individual contributor in Construction on their own teams and is beginning to provide leadership within their own teams. This person has worked multiple completed projects and has experience in each of the basic software development lifecycle steps needed to release a product.

The expected time to complete this level is again 1-2 years, but closer to 2 years than to 1.

Figure 4 Requirements to complete Level 2 on the Software Architect Path

	Knowledge Area										
Capability Level	Configuration Management	Construction	Design	Foundations	Maintenance	Management	Models and Methods	Process	Quality	Requirements	Testing
Introductory	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Competence	●	✓	●								✓
Leadership		●									

Level Proficiency Goals

Upon completion of this level, this person will have added proficiencies including proficiency with a version control tool and build process, API design, class, program, and routine design, program design, refactoring, release management and supporting tooling, expertise with at least four programming languages / technologies, expertise with interface/API implementation techniques —and collectively has demonstrated a track record of significant contribution to Construction and Design on their projects.

Overall, this person will have the capability to be recognized as a coach in Construction.

Reading

Competence

- Configuration Management Best Practices: Practical Methods that Work in the Real World, Aiello and Sachs (272 pages, Analytical) (CNFG)
- Version Control by Example, Rick Sink (288 pages, Inspectional) (CNFG)
- Head First Design Patterns, Freeman et all (694 pages, Inspectional) (DSGN)

Leadership

- Clean Code: A Handbook of Agile Software Craftsmanship, Robert Martin (464 pages, Syntopical) (CNST)
- The Pragmatic Programmer: From Journeyman to Master, Andrew Hunt and David Thomas (352 pages, Syntopical) (CNST)

Training

Competence

- DevOps Essentials (~1 day) (CNFG)
- Design in Depth (~2 days) (DSGN)
- Design Patterns in Depth (~1 day) (DSGN)
- One or more approved design techniques course from Pluralsight or LinkedIn Learning (or equivalent) (~1 day) (DSGN)

Work Activities

Competence

Configuration Management-Focused

- Define and implement a branching strategy for a small project (CNFG)
- Lead a review and discussion of project artifacts related to Configuration Management (CNFG)
- Lead the release process for a significant project (CNFG)
- Participate as a reviewer in a formal review of one or more project artifacts in Configuration Management (CNFG)
- Participate in the creation and execution of a project release process (CNFG)

- Participate in the creation of a project's build process and infrastructure (CNFG)
- Participate in the organization, identification, and revision control of a project's artifacts (CNFG)

Design-Focused

- Apply normalization rules to a database design (and then denormalize it, as necessary) (DSGN)
- Contribute to a design process and artifact creation (DSGN)
- Critique multiple programs (including open source) for their design (DSGN)
- Design and implement a database (DSGN)
- Design multiple APIs (DSGN)
- Design multiple programs (DSGN)
- Design numerous classes (DSGN)
- Design numerous low-level routines (DSGN)
- Lead a review and discussion of project artifacts related to Design (DSGN)
- Participate as a reviewer in a formal review of one or more project artifacts in Design (DSGN)
- Participate in design of a significant program (DSGN)
- Refactor multiple classes (DSGN)

Leadership

- Act as a consultant / coach / mentor to other project teams in Construction (CNST)
- Create and deliver a presentation on the collaborative construction approach used for a major project (CNST)
- Create and deliver a presentation on the debugging approach used for a major project (CNST)
- Implement a coding or construction standard (CNST)
- Lead 3 or more collaborative construction efforts (CNST)
- Lead a performance profiling / tuning effort (CNST)
- Lead code review activities for a major project (CNST)
- Lead collaborative construction activities for a significant project (CNST)
- Lead design review activities for a major project (CNST)
- Lead work in Construction for a major system (CNST)
- Provide consulting / coaching / mentoring in Construction within a team (CNST)

- Teach an evening or weekend class or make a conference presentation in Construction (CNST)
- Train a team in the use of code reading reviews (CNST)
- Train a team in the use of pair programming (CNST)

Level 3: "Technical Lead"

By completing work in Level 3 the professional will achieve Competence in Maintenance and Quality and Leadership in Design. At this level this path diverges significantly from other paths except for the Software Technical Manager path. Before committing to this level we advise the professional to carefully consider whether he or she really wants to pursue Architecture or Technical Management, or whether he or she might prefer a different path.

The professional who chooses to complete Level 3 will accumulate a series of "wins" during his or her participation in all aspects of small and large projects and will be essential to those projects' successes. This person will acquire a track record of consistently rendering clear technical judgment and routinely considering project-level issues. This person often provides leadership to others within their own teams in the areas of Construction and Testing and is beginning to provide leadership to the larger organization.

The expected time to complete this level is about 2 years.

Figure 5 Requirements to complete Level 3 on the Software Architect Path

	Knowledge Area										
Capability Level	Configuration Management	Construction	Design	Foundations	Maintenance	Management	Models and Methods	Process	Quality	Requirements	Testing
Introductory	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Competence	✓	✓	✓		●				●		✓
Leadership		✓	●								

Level Proficiency Goals

Upon completion of this level, this person will have added proficiencies including leading Agile retrospectives, team-level quality improvement, proficiency with debugging, refactoring, and maintaining an existing system.

This professional will have developed expertise (not just proficiency) in relational database design; system and architectural design techniques; user interface design techniques; relational database design; system and architectural design techniques; and system design—and has accumulated a track record of significant work in Design on multiple, diverse projects.

Overall, this person will have the capability to be recognized as a coach in Design as well as in Construction.

Reading

Competence

- Agile Retrospectives, Derby and Larsen (200 pages, Inspectional) (QUAL)
- Competitive Engineering, Tom Gilb, Specification Quality Control Chapter (40 pages, Analytical) (QUAL)
- Handbook of Software Quality 4th Edition, Chapters 1, 2, 4, 7, 12, 15, G. Gordon Schulmeyer (151 pages, Analytical) (QUAL)
- Peer Reviews in Software, Chapters 3-4, Karl Wiegers (29 pages, Analytical) (QUAL)

Leadership

- Clean Architecture: A Craftsman's Guide to Software Structure and Design, Robert C. Martin (432 pages, Syntopical) (DSGN)
- Domain Driven Design, Eric Evans (560 pages, Inspectional, as part of a Syntopical Reading set) (DSGN)
- Object Oriented Design Heuristics, Arthur Riel (400 pages, Inspectional, as part of a Syntopical Reading set) (DSGN)
- Software Architecture in Practice 3rd Edition, Bass, Clements, and Kazman (646 pages, Syntopical) (DSGN)

Training

Competence

- Total Project Quality (~1 day) (QUAL)
- Understanding Software Project Lecture Series - "Defect" Lectures (~3 hours) (QUAL)

Leadership

- API Design (~3 days) (DSGN)
- Model Based Design (~2 days) (DSGN)

Work Activities

Competence

Maintenance-Focused

- Complete multiple perfective fixes for an existing system (MNTC)
- Complete multiple small enhancements for an existing system (MNTC)
- Lead a review and discussion of project artifacts related to Maintenance (MNTC)
- Lead debugging and resolution of a production problem in an existing system (MNTC)
- Participate as a reviewer in a formal review of one or more project artifacts in Maintenance (MNTC)
- Participate in backlog refinement and/or Kanban of work on an existing system (MNTC)
- Participate in creation of an install application (MNTC)
- Participate in migration of a software system (MNTC)

Quality-Focused

- Assist in a Root Cause Analysis (RCA) session (QUAL)
- Assist in creation of quality metrics for a significant project (QUAL)
- Establish release quality criteria (QUAL)
- Facilitate multiple retrospectives (PRCS, MGMT, QUAL)
- Lead a quality tracking effort for a significant project (QUAL)
- Lead a review and discussion of project artifacts related to Quality (QUAL)
- Lead a team in the creation of a Definition of Done (or equivalent) (QUAL)
- Participate as a reviewer in a formal review of one or more project artifacts in Quality (QUAL)
- Participate in creation of a quality plan (QUAL)
- Participate in defect management for a significant project (QUAL)
- Participate in the identification of non-functional requirements using Planguage (QUAL)
- Review and classify defects by activity introduced (requirements, design, construction, testing) (QUAL)

Leadership

- Act as a consultant / coach / mentor to other project teams in Design (DSGN)
- Create a design for a new system (DSGN)
- Create a system architecture for a significant system (DSGN)

- Create a user-interface design (DSGN)
- Create an artifact for a major system (e.g. API Design, UI Design, Database Design, Architecture) (DSGN)
- Create and deliver a presentation on the design approach for a major system (DSGN)
- Lead a database design effort (DSGN)
- Lead a major design adaption effort for an existing system (DSGN)
- Lead a model-based design effort for a major system (DSGN)
- Lead creation of unit testing guidelines for a significant system (DSGN, TEST)
- Lead design activities for a significant project (DSGN)
- Lead design of a system for testability (DSGN, TEST)
- Lead refactoring of an existing system with automated tests (DSGN)
- Lead the design of a significant project (DSGN)
- Lead the design of APIs for a major system (DSGN)
- Lead work in Design for a major system (DSGN)
- Participate in the creation of a system architecture for a significant system (DSGN)
- Provide consulting / coaching / mentoring in Design within a team (DSGN)
- Teach an evening or weekend class or make a conference presentation in Design (DSGN)

Level 4: "Associate Architect"

Level 3 on this career path is considered to be an acceptable and reasonable plateau level, and not every professional will choose to advance to Level 4. A Level 3 professional on this path is capable of providing significant technical leadership in support of business objectives. The difference between Level 3 and Level 4 is the acquisition of significant non-coding-related capabilities needed to develop and support the holistic technical judgment that is characteristic of a true software architect.

Upon completing work to achieve Level 4, the professional will achieve Competence in Models and Methods and Requirements, and Leadership in Testing. The Level 4 professional is now providing leadership in the full set of skills needed for a top developer, including in the areas of Construction, Testing, and Design.

This person has become a champion who can consider both internal and external aspects of a project and ensures they are handled correctly and with consistently sound judgment. This person takes total ownership for all aspects of his or her project and makes many unique contributions. This person's decisions have a significant impact on their project's success. This person routinely provides leadership within their own teams and is recognized as a a major technical resource to others in their organization. As with the other levels, the expected time to complete the work in Level 4 is about 2 years.

Figure 6 shows the requirements for the professional to complete Level 4.

Figure 6 Requirements to complete Level 4 on the Software Architect Path

	Knowledge Area										
Capability Level	Configuration Management	Construction	Design	Foundations	Maintenance	Management	Models and Methods	Process	Quality	Requirements	Testing
Introductory	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Competence	✓	✓	✓		✓		●		✓	●	✓
Leadership		✓	✓								●

Level Proficiency Goals

Upon completion of this level, this person will have added proficiency in at least two modeling techniques, at least two requirements elicitation techniques, at least two requirements specification techniques—and overall will have established a track record of significant contributions in Requirements and Methods.

Moving beyond proficiency, this person will have added expertise in test automation; boundary and equivalency test case techniques; test design techniques; and test driven development.

This person is now recognized as a coach in Construction, Design, and Testing, and because of this combination of leadership abilities this person is also recognized as a leading software development contributor overall.

Reading

Competence

- Discussion of the Method: Conducting the Engineer's Approach to Problem Solving, Billy Vaughn Koen (276 pages) (MTHD)
- Software Requirements 3rd Edition, Parts I-II, Karl Wiegers (382 pages, Analytical) (RQMT)
- User Stories Applied, Chapters 2, 6, 7, Mike Cohn (28 pages, Analytical) (RQMT)

Leadership

- A Practitioner's Guide to Software Test Design, Lee Copeland (300 pages, Syntopical) (TEST)
- Developer Testing: Building Quality into Software, Alexander Tarlinder (352 pages, Syntopical) (TEST)
- Technology specific book such as Test Driven Development for Embedded C, James Grenning; Test Driven Java Development, Farcic and Garcia (356, 286 pages, Syntopical) (TEST)
- Tool specific book such as FIT for Developing Software, Mugridge and Cunningham; The Cucumber Book for Java: Behavior-Driven Development for Testers and Developers (336, 384 pages, Inspectional, as part of a Syntopical Reading set) (TEST)

Training

Competence

- Model-Based Design (~2 days) (MTHD)
- Model-Based Requirements (~2 days) (MTHD)
- Agile Requirements in Depth OR Requirements Boot Camp (~1 day) (RQMT, RQMT)
- Agile Requirements Modeling OR Model Based Requirements (~1 day) (RQMT, RQMT)

Work Activities

Competence

Models and Methods-Focused

- Evaluate the way in which design is represented on a project, focusing on modeling approach and notation and propose improvements (MTHD)
- Evaluate the way in which requirements are represented on a project, focusing on modeling approach and notation and propose improvements (MTHD)
- Lead a review and discussion of project artifacts related to Methods (MTHD)
- Participate as a reviewer in a formal review of one or more project artifacts in Methods (MTHD)
- Participate in creating a structural, behavioral, or information model (MTHD)
- Participate in using heuristic methods (MTHD)

Requirements-Focused

- Contribute to a requirements discovery session (story writing workshop, story mapping workshop, JAD session, etc.) (RQMT)
- Lead a review and discussion of project artifacts related to Requirements (RQMT)
- Lead multiple refinement meetings (or equivalent) (RQMT)
- Participate as a reviewer in a formal review of one or more project artifacts in Requirements (RQMT)
- Select and implement a requirements approach (use cases, user stories with acceptance criteria, UML models, etc.) for a project (RQMT)
- Specify functional and non-functional requirements for a project using a specification technique (user stories with acceptance criteria, etc.) (RQMT)

Leadership

- Act as a consultant / coach / mentor to other project teams in Testing (TEST)
- Create automated tests on a significant project (TEST)
- Create automated unit tests for multiple modules using TDD (TEST)
- Lead development of an automated test system for a significant project (TEST)
- Lead test strategy and test planning for a significant project (TEST)
- Participate in compatibility testing for a significant project (TEST)
- Participate in performance testing for a significant project (TEST)
- Participate in scalability testing for a significant project (TEST)

- Participate in stress testing for a significant project (TEST)
- Provide consulting / coaching / mentoring in Testing within a team (TEST)
- Teach an evening or weekend class or make a conference presentation in Testing (TEST)

Level 5: "Software Architect"

Level 4 on this career path is considered to be an acceptable and reasonable plateau level, and most professionals will not choose to advance to Level 5. The difference between Level 4 and Level 5 is the development of end-to-end project understanding needed to develop additional support for broad technical judgment and deep leadership. Career path variations that focus more on business capabilities than engineering are discussed in the "Customization and Adaptation" section of this document.

By completing Level 5 the professional will achieve Leadership in Quality and Requirements. The Level 5 professional is now providing leadership in the full set of skills needed for a Software Architect, including in the areas of Construction, Testing, Design, Quality, and Requirements.

This person has become a champion who fully considers internal and external aspects of sets of projects, the effects of a portfolio of projects on the company's technical and business strategy, and the success of the portfolio of projects with customers. This person takes ownership of the company's technical strategy and makes unique contributions to that. This person routinely provides broad technical leadership throughout the organization, and this person's technical decisions have a significant impact on the company's success. The expected time to complete Level 5 is about 2 years.

Figure 7 shows the requirements for the professional to complete Level 5.

Figure 7 Requirements to complete Level 5 on the Architect Path

	Knowledge Area										
Capability Level	Configuration Management	Construction	Design	Foundations	Maintenance	Management	Models and Methods	Process	Quality	Requirements	Testing
Introductory	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Competence	✓	✓	✓	●	✓		✓		✓	✓	✓
Leadership		✓	✓						●	●	✓

Level Proficiency Goals

Upon completion of this level, this person will have added proficiency in economic decision-making; creating, reviewing and evaluating business cases; and performing basic economic analysis of inaccuracy, risk, and uncertainty.

This person will have added expertise in organization-level quality improvement, defect management, root cause analysis, and a full spectrum of requirements activities including elicitation, modeling, specification, and facilitation.

This person is now recognized as a coach in Quality and Requirements, and because of this combination of leadership abilities this person is also recognized as a company-level technical leader.

Reading

Competence

- Return on Software, Steve Tockey, Chapter 1-5, 7-11, 18-20 (187 pages, Analytical) (FNDN)

Leadership

Quality-Focused

- Competitive Engineering, Tom Gilb (480 pages, Inspectional, as part of a Syntopical Reading set) (QUAL, RQMT)
- Measuring Defect Potentials and Defect Removal Efficiency, Capers Jones (3 pages, Syntopical) (PRCS, QUAL)
- Quality Software Management Vol 1, Gerald Weinberg (336 pages, Syntopical) (QUAL)
- The Economics of Software Quality, Capers-Jones and Bonsignour (624 pages, Syntopical) (QUAL)

Requirements-Focused

- BDD in Action, John Ferguson Smart (384 pages, Syntopical) (RQMT)
- Competitive Engineering, Tom Gilb (480 pages, Inspectional, as part of a Syntopical Reading set) (QUAL, RQMT)
- Mastering the Requirements Process 3rd Edition, Chapters 7, 12, Robertson and Robertson (52 pages, Syntopical) (RQMT)
- Mastering the Requirements Process 3rd Edition, Robertson and Robertson (768 pages, Inspectional, as part of a Syntopical Reading set) (RQMT)
- Requirements Engineering: From System Goals to UML Models to Software Specifications, Part II, Axel van Lamsweerde (247 pages, Syntopical) (RQMT)

- User Story Mapping: Discover the Whole Story, Build the Right Product, Jeff Patton (324 pages, Syntopical) (RQMT)

Training

Competence

- Engineering Economics Fundamentals (~2 hours) (FNDN)

Leadership

- Take an instructor-led facilitation seminar or class (~2 days) (PRCS, MGMT, RQMT)

Work Activities

Competence

- Create a project's business case or similar economic analysis (FNDN)
- Perform a break-even analysis or optimization analysis (FNDN)
- Perform an economic analysis involving allowance for inaccuracy, risk, or uncertainty (FNDN)

Leadership

Quality-Focused

- Act as a consultant / coach / mentor to other project teams in Quality (QUAL)
- Create a quality plan for a large and a small project (QUAL)
- Create artifacts related to a major system (e.g. quality plan, metrics, root cause analysis) (QUAL)
- Lead a quality improvement initiative (QUAL)
- Lead a Root Cause Analysis (RCA) session (QUAL)
- Lead a root-cause analysis effort (QUAL)
- Lead an organizational change (QUAL)
- Lead defect management for a significant project (QUAL)
- Lead quality activities for a significant project (QUAL)
- Lead quality tracking efforts on a significant project (QUAL)
- Lead work in Quality for a major system (QUAL)
- Provide consulting / coaching / mentoring in Quality within a team (QUAL)

- Teach an evening or weekend class or make a conference presentation in Quality (QUAL)

Requirements-Focused

- Act as a consultant / coach / mentor to other project teams in Requirements (RQMT)
- Create a domain model on a medium to large project (RQMT)
- Create a user story map (RQMT)
- Create an activity model for a medium to large project (RQMT)
- Create low-fi UI prototype to uncover requirements (RQMT)
- Create use cases for a for a medium to large project (RQMT)
- Facilitate a workshop in story mapping, story writing, or other related area (RQMT)
- Lead a requirements discovery session (RQMT)
- Lead user story mapping effort for a significant project (RQMT)
- Provide consulting / coaching / mentoring in Requirements within a team (RQMT)
- Teach an evening or weekend class or make a conference presentation in Requirements (RQMT)
- Write BDD style requirements (RQMT)
- Write Planguage style requirements (RQMT)
- Write user stories and acceptance criteria (RQMT)

Professional Development Plans

Each career path in Construx's PDL is designed to take several years to complete. The scale of work required to progress through Construx's career paths can seem daunting, despite the fact that the workload in Construx's PDL is actually easily manageable—with an expected reading pace of only about 20 pages per week.

To make the workload manageable and to ensure consistent progress, Construx recommends creation of Professional Development Plans (PDP) that define shorter-term goals and track shorter-term progress, guided by the longer-term goal of a path-level transition. The typical structure looks like this:

Overall Goal *Finish work for Level 2 within 18 months* (This is just an example and can be tailored by the individual.)

Next 3-6 months Weekly schedule and goals for specific reading, training, and work activities. The professional can then track their own progress closely and ensure they are on track to meet their self-imposed professional development goals. As described above, the expected workload is typically about 20 pages of reading per week, plus training and directed work experiences.

6-12 months Monthly schedule for large-scale work completion, based on completing cells in the PDM (e.g., "April: Complete reading requirements for competence in Testing; complete training requirements for competence in Construction."

13-24 months A high level path is defined focusing on large-scale work completion, which is typically easiest if it focuses on completing cells in the PDM.

Most professionals will find that as they approach their detailed 3-6 month planning, they will want to depart somewhat from trying to follow the PDM strictly cell-by-cell. This is acceptable and expected.

In many organizations it works well to align PDP timing with the organization's performance review cycle. If review cycles are more than six months apart, however, we recommend setting short-term goals at 3-6-month intervals.

Customization and Adaptation

The fact that Construx's career paths are based on the PDM allows for customization of career paths without losing the structure and discipline associated with the default path.

Addition of Knowledge Areas

The 11 KAs in Construx's PDM are focused on software development best practices. We intentionally do not include technology-specific or business-specific KAs. In our work with companies implementing software career pathing, however, we have found that most companies will add technology-specific and/or business-area specific KAs, while keeping Construx's PDL as the overall professional development framework. This allows for company-specific customized PDMs that look like this:

Figure 8 Customized requirements for the Software Architect Path—Example

	Knowledge Area											
Capability Level	Configuration Management	Construction	Design	Foundations	Maintenance	Management	Models and Methods	Process	Quality	Requirements	Testing	**Business Area**
Introductory	●	●	●	●	●	●	●	●	●	●	●	●
Competence	●	●	●	●	●		●		●	●	●	●
Leadership		●	●						●	●	●	●

Substitutions for Specific PDL Requirements

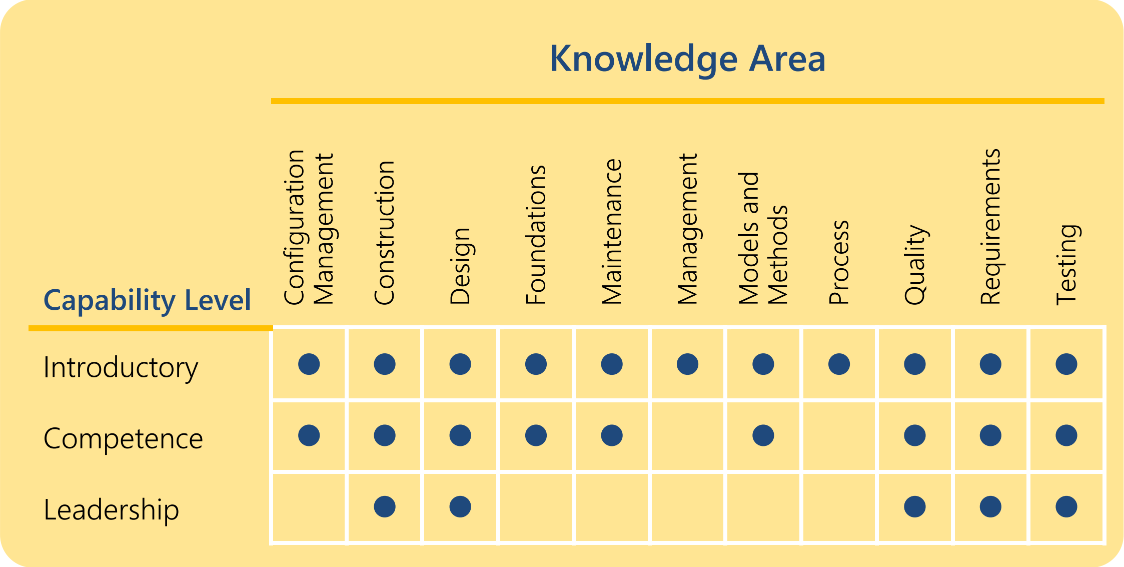
Professionals will often find that they have completed reading, training, or work experience comparable to one of the PDL's specific requirements. For example, a professional might have taken a classroom training session in Scrum that is comparable to the self-paced Scrum training requirement in the PDL. In that case, the professional can check that item as complete if the training was substantially similar.

Likewise, if comparable reading has been completed, a reading requirement can be declared to be complete. One caution with reading: Construx's PDL focuses on *book-length* reading. We believe that the topics in the PDL require a book-length exposition for meaningful understanding. Reading a blog article, or a few blog articles, is typically not an acceptable substitute for one of the books defined in the PDL.

Changes in Emphasis in a Career Path

The emphasis of a pre-defined career path can be changed through changes in the Competence and Leadership requirements without losing the structure and discipline associated with the path. For the most part, it is acceptable to make one-for-one changes in PDL requirements. For example, the nominal path for a Software Architect looks like this:

Figure 9 Full requirements for the Software Architect Path



	Knowledge Area										
Capability Level	Configuration Management	Construction	Design	Foundations	Maintenance	Management	Models and Methods	Process	Quality	Requirements	Testing
Introductory	●	●	●	●	●	●	●	●	●	●	●
Competence	●	●	●	●	●		●		●	●	●
Leadership		●	●						●	●	●

For a professional who wanted a more pure-technical path, they could exchange the Leadership requirement in Requirements for Leadership in Models and Methods, resulting in requirements that look like this:

Figure 10 Alternate requirements for the Software Architect Path

	Knowledge Area										
Capability Level	Configuration Management	Construction	Design	Foundations	Maintenance	Management	Models and Methods	Process	Quality	Requirements	Testing
Introductory	●	●	●	●	●	●	●	●	●	●	●
Competence	●	●	●	●	●		●		●	●	●
Leadership		●	●				🟢		●	✕	●

Similarly, a person who wanted a more business-oriented path could trade Leadership in Testing for Leadership in Management, like this:

Figure 11 Alternate requirements for the Software Architect Path

	Knowledge Area										
Capability Level	Configuration Management	Construction	Design	Foundations	Maintenance	Management	Models and Methods	Process	Quality	Requirements	Testing
Introductory	●	●	●	●	●	●	●	●	●	●	●
Competence	●	●	●	●	●	🟢	●		●	●	●
Leadership		●	●			🟢			●	●	✕

This path could also be augmented with work in a company-specific "Business Area" KA, as described earlier.

Construx's PDL has been defined in such a way that each individual professional can pursue their interests and tailor the PDL to their specific goals while retaining the beneficial structure of the ladder and maintaining approximately the same professional development workload.

Summary

Construx's Software Professional Career Paths provide well-defined programs for long-term skills development in software professionals. The structure and details of the programs are based on industry standards and Construx's experience working with leading companies worldwide for more than 20 years. The Software Architect program described in this white paper is deep and comprehensive. Total time required to progress through the set of five Career Path Levels is 8-10 years. The individual who completes the career path defined in this white paper will be a highly skilled professional capable of providing leadership to their teams and organization.

References

"Agile Technical Coach Career Path," Construx Software, 2018.

"Career Paths for Software Professionals", Construx Software, 2018.

"How to Read a Technical Article," Steve McConnell, *IEEE Software*, Nov/Dec 1998.

"PDL Implementation Guide", Construx Software, 2018.

"Product Owner Career Path," Construx Software, 2018.

Professional Software Development, Steve McConnell, Addison-Wesley, 2004.

"Software Quality Manager Career Path," Construx Software, 2018.

"Software Technical Manager Career Path," Construx Software, 2018.

Contributors

Steve McConnell, CEO and Chief Software Engineer

steve.mcconnell@construx.com
+1(425) 636-0107

Jenny Stuart, VP Consulting

jenny.stuart@construx.com
+1(425) 636-0108

Construx

Construx Software is the market leader in software development best practices training and consulting. Construx was founded in 1996 by Steve McConnell, respected author and thought leader on software development best practices. Steve's books *Code Complete*, *Rapid Development*, and other titles are some of the most accessible books on software development with more than a million copies in print in 20 languages.

Steve's passion for advancing the art and science of software engineering is shared by Construx's team of seasoned consultants. Their depth of knowledge and expertise has helped hundreds of companies solve their software challenges by identifying and adopting practices that have been proven to produce high quality software—faster, and with greater predictability. For more information about Construx's support for software development best practices, contact us at consulting@construx.com, or call us at +1(866) 296-6300.