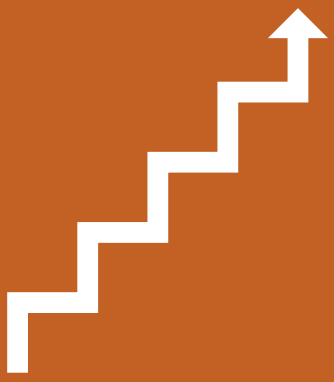
Continual Learning



Building a DevOps-Inspired Career Track

IT REVOLUTION
DevOps Enterprise Forum 2017

Continual Learning

Building a DevOps-Inspired Career Track

IT REVOLUTION
DevOps Enterprise Forum 2017
Portland, Oregon

25 NW 23rd Pl
Suite 6314
Portland, OR 97210

Continual Learning:
Building a DevOps-Inspired Career Track

First Edition
Produced in the United States of America
10 9 8 7 6 5 4 3 2 1

Cover design and interior by Devon Smith

For further information about IT Revolution, these and other publications,
special discounts for bulk book purchases, or for information on booking authors
for an event, please visit our website at www.ITRevolution.com.

PREFACE

In April of this year, we at IT Revolution had the pleasure of hosting technology leaders and experts from across the DevOps Enterprise community at the DevOps Enterprise Forum event in Portland, Oregon. The Forum's ongoing goal is to create written guidance to overcome the top obstacles facing the DevOps Enterprise community.

Each year at the Forum, the topics covered have included organizational culture and change management, architecture and technical practices, metrics, integrating and achieving information security and compliance objectives, creating business cases for automated testing, organizational design, and many more.

For the first two years, we organized the participants into large teams that worked on a small number of broad topics. However, this year, we shifted our approach in two ways—first, we invited a core group of past participants to propose topics they would like to work on and second, we asked them to narrow their topics so that they could have "nearly shippable" artifacts by the end of the second day. The result was more teams working on more problems with more written guidance.

After the Forum concluded, the groups spent the next eight weeks working together to complete and refine the work they started together. The results can be found in this year's collection of Forum papers. I hope you will agree that the smaller teams and reduced scope of the guidance benefits both the teams as well as the reader.

IT Revolution is proud to share the outcomes of the hard work, dedication, and collaboration of the amazing group of people from the 2017 DevOps Enterprise Forum. Our hope is that you will gain valuable insight into DevOps as a practice.

—Gene Kim
June 2017
Portland, Oregon

DevOps is not a goal, but a never-ending process
of continual improvement.

—**Jez Humble**

We start from the presumption that our people
are talented and want to contribute. We accept that,
without meaning to, our company is stifling that talent
in myriad unseen ways. Finally, we try to identify
those impediments and fix them.

—**Ed Catmull**

INTRODUCTION

Many organizations list a skills gap as one of their primary obstacles to achieving and sustaining a high-functioning DevOps-based organization. This paper attempts to provide practical guidance to organizations on developing, recruiting, and retaining DevOps talent.

Two-thirds of CIOs say they believe a lack of talent has become one of the key limiting factors for organizations who want to scale-up DevOps on a broad level.[1] Employers need to recruit and retain more "T-shaped talent"[2] (individuals who have both depth in a specialty but also breadth across the spectrum, often visually represented by a T shape) to deliver in a DevOps manner. Companies need managers and leaders who understand practices of Lean/Agile product management instead of the traditional project and process management focus. For the employees, the best way to learn and gain these cross functional skills is to work in an existing DevOps organization.

A DevOps organization focuses on a different set of skills:

- DevOps involves different ways of thinking than traditional IT, including flow, systems thinking, growth mindset, seeking feedback cycles, and the expertise to automate.
- DevOps organizations work collaboratively on cross-functional teams versus single function teams.
- DevOps demands T-shaped skill sets: individuals who have both depth in a speciality but also breadth across the spectrum. This is often called the "full stack engineer."
- Finally, in DevOps organizations the management and leadership shift from the traditional schools of project and process management toward the Lean/Agile ideas of product management.[3]

---

[1] Peter Borner, "Managing the DevOps Talent Challenge," PrecipiceLLP.com, http://www.percipiencellp.com/managing-the-devops-talent-challenge/.

[2] The term T-shaped was first described by David Guest in 1991 in the article "The hunt is on for the Renaissance Man of computing," in *The Independent*, September 17, 1991.

[3] Project and process management remain important components of delivering digital systems; the challenge is in understanding their limitations and correct use.

The current educational model for IT talent does not provide adequate training for DevOps skills. Therefore, employers often need to implement internal DevOps training and career programs of their own. This paper seeks to provide a set of starter principles, guidance, and tools to help companies establish a DevOps inspired career track and training program.

This paper also serves as a call to action for the industry to help advise and advocate for more DevOps and Lean/Agile concepts to be included in the curriculum of traditional educational providers. Academic partners for both four-year degrees and adult education have not widely embraced teaching updated principles and current skill sets. As a community, we have a role to play in advocating for better educational programs for the workforce.

CAREER PATH

> "The illiterate of the 21st century will not be those who cannot read and write but those who cannot learn, unlearn, and relearn."
>
> —**Alvin Toffler**

An information technology career path involves the progressive development of an individual throughout their career. While there could be many steps along this path, for simplicity the guidance model we will use will provide principles and guidance on career steps based on three levels, starting with an entry level role to a mid-level master to an expert level leader. Additionally, we recommend that the path include a branch for technology professionals who desire to take on people management responsibilities while also providing a clear path those who wish to advance their career in a technical leadership direction.

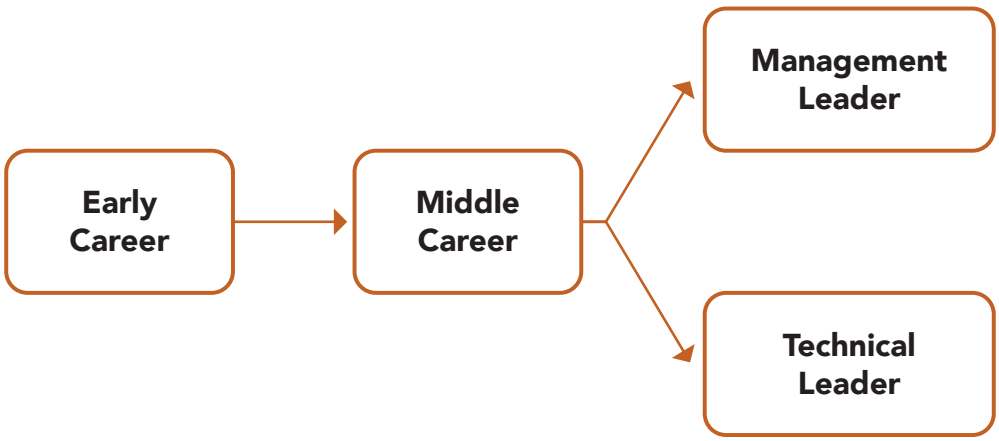


Fig. 1: Three-Level IT Career Path

Principles, Recommendations, and Tools

Based on the three-level approach above, we discovered principles, guidance recommendations, and tools that can be helpful to promote a healthy DevOps career. We explore these areas for each of the three levels.

- **Principles:** The principles provide direction on how each category of learner should approach the learning model as they navigate the career progression ladder.
- **Guidance:** Guidance will offer ideal patterns for learners and leaders to apply to their individual journeys within an enterprise.
- **Tools:** Tools will provide some example templates to help guide learners and leaders as they assess their current skills state, identify gaps and opportunities, and take a structured approach to enable their transformation.

EARLY CAREER

Learners

IT and software engineering practices are moving from a bespoke model where enterprises built custom capabilities and skills that were not always transferable to a model built on open platforms, tooling, and patterns. Interns, college graduates, junior engineers, and entry-level IT professionals looking to begin their career in a DevOps model should consider the more generalist nature of their roles. In addition to their formal training/education, which enables them to build deep expertise in one function, they should focus on the breadth of functions necessary to make a full-stack team successful.

Teachers/Coaches/Leaders

Leaders responsible for building and sustaining high-performance teams should ensure two key aspects: (1) ensure early engagement with intake pipelines (college and intern programs, early prep); (2) establish a growth mindset learning pattern for new employees.

Principles

- **Stack Fallacy:** Education about core computer science concepts is necessary, but not sufficient. To enable talent to have the right skill and to operate efficiently, the specifics of process, tooling, and methodologies needed for software development in an enterprise context is a fundamental necessity.
- **Curiosity:** Leaders should facilitate experimentation, provide the time and freedom to learn new concepts and tools, and encourage risk-taking as a core part of the job.
- **Collaboration:** While focused effort is often required to solve problems, system-wide optimization requires reaching out across teams and bridging functional and domain expertise. Leaders should encourage cross-discipline forums and teamwork.
- **Optimism:** Hope can help provide an oxygen-rich environment to ignite a culture of innovation, experimentation, and motivation. Leaders should set the example by demonstrating and encouraging positive perspectives.

Guidance

- **Drive Experiential Learning:** Leaders and educators should encourage a hands-on approach to training and development.

- **Prioritize New Skills:** Technological advances are accelerating and require a workforce that is able to adapt quickly. Leaders and educators need to help prepare the workforce by proactively incorporating training to develop new skills to master nascent and emerging tech.
- **Employ Scientific Method for Problem Solving:** Often, technology workers will be tempted to solve problems in haphazard ways, using instinct or a "gut feeling" approach. This can result in a shotgun approach of uncontrolled or unmeasured changes attempting to solve the problem. Leaders and educators should encourage workers to approach problems in a methodical, deliberate, and iterative fashion by using the scientific method of systematic observation, measurement, experimentation, testing, and modification.

Tools

Role-Based Skills Matrix

Leaders and educators can use a skills matrix to measure and assess the skills, knowledge, and interest of individuals on a team for a particular role, helping identify opportunities for placement and training. It is generally structured as a table that lists the various skills and knowledge areas required for a particular role matrixed against the skills and interest of individual members of a team.

Role-Based Skills Matrix Example

	Ann		Brian		Carrie		David	
	Proficiency	Interest	Proficiency	Interest	Proficiency	Interest	Proficiency	Interest
Systems Architecture	5	1	1	0	2	1	3	1
Java Development	2	0	5	1	3	0	3	0
Design Patterns	3	0	3	0	5	1	1	1
Technical Writing	2	1	1	0	1	0	1	1
Graphic Design	5	0	1	0	2	0	4	1
Teaching	2	0	1	0	5	1	1	0
Scrum Master	1	1	5	1	4	0	1	0

Proficiency: 1 = Basic; 5 = Expert / Interest: 0 = No interest in apply this skill or knowledge; 1 = Interest

MIDDLE CAREER

Learners

IT professionals who have mastered the principles listed above may have reached a point in their career where they are very well established in their field, are sought after as an expert, and are productive in delivering business value. However, it is at this point that individuals must make a concerted effort to stay current and resist the propensity to become vertically specialized without horizontal, cross-discipline collaboration.

Teachers/Coaches/Leaders

Leaders responsible for building and sustaining high-performance teams should:

- Reinforce continual learning and a growth mindset.
- Develop individuals to expand their mastery horizontally across new disciplines.
- Give individuals the ability to move horizontally among roles/specialties in order to gain more breadth toward the T-shaped model.

Principles

- **Adult Learning**: Even masters must continually learn.
- **T-Shaped Talent**: Deep vertical expertise and specialization areas should form as part of an individual's mastery, but individuals must also be mindful to develop a horizontal ability to collaborate across disciplines and with many other experts in other areas.
- **Growth versus Fixed Mindset**: It is important that a DevOps professional maintain a growth mindset.

Guidance

- **Emphasize Systems Thinking**: Practitioners at this stage need to understand concepts such as optimizing whole versus parts, non-linearity, variation, feedback, and complexity. It is especially important to educate on the appropriate use of process management and when it is—and is not—appropriate to seek reduction in variability.

- **Prioritize Clinical Practice**: As suggested with the CALMS acronym (culture, automation, learning, measurement, sharing), measuring and analysing DevOps and digital operations is essential. Practitioners at this stage need fundamental statistics (including understanding various distributions) as well as applications, such as measuring the loading of queues and their throughput.

Tools

T-Shaped Talent Development Skills Family Assessment Model

This table lists the major skills required for a full stack team with multiple functional team specializations. Leaders and individual team members can use this to assess areas to develop horizontally.

T-Shaped Talent Development: Skills Family Assessment Model

	Product Owner	Scrum Master	Code	DB	DevSecOps	CloudOps	Security
Capabilities	Develop-ing project charters	Understand Scrum	Understand 12-factor applications	Large scale distributed DB	Design a continuous delivery pipeline	Design cloud formation templates	Build a threat-assessment model
	Managing a project backlog	Onboard a team into scrum model	Know how to decompose larger appli-cations	NoSQL DB	Enable automated testing for function, performance, and security	Optimize costs for AWS services	Assess data protection, assess management needs
	Prioritizing deliverables	Facilitate effective meetings	Design for elasticity and resilience	Design multi-site replication	Provide continuous measurement and improve-ment	Evaluate and use cloud native services	Build real-time automated threat detection and reaction frameworks
	Maintaining a RACI	Scale Scrum to large, multi-region projects	Reusable patterns and frameworks	DB changes via DevOps		Architect roll forward/back A/B environ-ments	Engineer security into CI/CD process
	Identifying velocity & driving planning	Continuous improvement	Open source	Virtualization as a service			

LEADERS

> Leadership requires passion. If you don't have it, get out now.
>
> —**Eric Schmidt & Jonathan Rosenberg**

Learners

IT professionals who have adopted the middle career principles and guidance and are looking to further expand their influence and impact will look to move into more of a leadership role. It is generally at this point that the choice between a management path or a technical leader path is considered. There are common principles to both paths, but there are also unique elements that we will consider.

Teachers/Coaches/Leaders

Leaders responsible for building and sustaining high-performance teams should:

- Train leaders to be teachers who are helping develop and mentor others.
- Model and reinforce a blameless culture that sees learning opportunity in failures.
- Establish and support a high-level technical individual contributor role (e.g., Technical Fellow or Distinguished Engineer) to help amplify the organization value and influence of the technical individual contributor track.

Principles

- **Leaders as Teachers:** Technical leaders strive to be educators in practice, using their advanced skills and experience to teach, mentor, and develop others.
- **Communication:** Technical leaders must be able to communicate with all levels across the organization. Their influence is especially critical for driving thought leadership and technical direction on products and standards.
- **Community Building:** Technical leaders exemplify cross-team collaboration, specifically leading efforts to build a community of technologies that encourage things like exploration, discussion, experimentation, and standardization. This involves both internal and external communities.

- **Blameless culture:** When honest mistakes or incidents occur, leaders reinforce a blameless culture by focusing on opportunities for learning and improvement. They reward honesty and drive no-fault incident reviews.
- **Technical Twinkle:** Technical leaders bring energy and technical enthusiasm to the team. They are known by their depth of knowledge and impressive capability, but they also illuminate and inspire other technologists that follow them.
- **Consistency:** Senior level technical leaders help provide stability, continuity, and historical context for their businesses and teams through their tenure of consistent record of successful leadership.

Guidance

- **Teach How to Coach:** Organization should adopt or develop programs that teach leaders how to coach others.
- **Teach and Reward Innovation:** Formal programs should be put into place to teach, recognize, and reward innovation.
- **Community Building:** Funding and support needs to be provided to leaders to launch and attend internal and external community building events, forums, and conferences.
- **Impactful Communication as Core Competence:** Leaders bring technical enthusiasm, inspiration, and direction that affects the entire organization. It is critical that these leaders are equipped with the necessary skills to deliver impactful communications in written and verbal form.

Tools

- College Curriculum
- Standards by IEEE & ACM

Summary Table

The following table provides a summary snapshot of the above-mentioned principles, guidance, and reference.

Summary Table

	Early Career		Mid Career		Leaders	
Principles	Grwoth vs. Fixed Mindset		Adult Learning		Leaders as Teachers	Leadership Training
	"Stack Fallacy"		"T-Shaped" Talent		Established Career Path/ Ladder	
Guidance	Drive Experimental Learning	Emphasize Systems Thinking	Prioritize Clinical Practice		Teach "How to Coach"	
	Prioritize New Skills				SME Focus Teach/Reward Innovation	Community Building
Reference & Tools	Role Based Skills Matrix				College Curriculum	Resource Clearinghouse
	Skills Gap				Standards– IEEE/ACM	

THE SUPPLY SIDE: WORKFORCE CREATION AND ONGOING DEVELOPMENT

> We now accept the fact that learning is a lifelong process of keeping abreast of change. And the most pressing task is to teach people how to learn.
>
> —**Peter Drucker**

There are a number of ways the digital/IT workforce is educated and trained, both pre-employment and ongoing learning, including traditional post-secondary education and continuing education in the form of professional frameworks and product-focused training.

But there appears to be a large and growing gap between industry practice and traditional four-year education. Traditional higher education in the computing, IT, and IS disciplines has the following issues, from a DevOps perspective:

- It is still too often based on waterfall precepts, and Agile is only covered generally, in its most limited sense of a software development methodology.
- It uncritically teaches project management and process management at a time when the role and appropriate use of both is in question.
- It does not cover product management in general, except in limited aspects, such as "requirements management" and "usability engineering." Concern for outcomes (i.e., "business" concerns) is deliberately out of scope in many computing courses.
- It does not cover operations, except in the IT specialty, which is generally considered a weak and poorly regarded degree option relative to computer science, software engineering, or information systems.
- It does not offer students effective practical experience with complex, realistic integrated systems. Version control is frequently not taught.

The traditional education sector should not be dismissed. It represents a large portion of the US and world economies and, although it responds slowly to feedback and has many challenges, it is still the means by which the majority of the workforce is created. Disruption may

occur, but today's DevOps hiring managers should not pin their hopes on non-traditional educational alternatives (e.g., boot camps), which represent only a small fraction of the educational system.

Many DevOps professionals are not aware of how computing-related curriculum (in the US, at least) is defined. A scoping report was produced in 2005[4] by the primary academic professional organizations (ACM, IEEE, AIS), which has resulted in a series of further reports defining curricula in disciplines such as computer science, software engineering, and management information systems, at two-year, four-year, and graduate levels.[5]

The entire structure of the disciplines is a sort of waterfall:

- Plan it with Information Systems
- Build it with Computer Science and Software Engineering
- Run it with Information Technology

DevOps and digital professionals seeking to influence college curricula and break down disciplinary barriers to students' developing a collaborative mindset will find themselves discussing this guidance with faculty. One recommendation of this paper is that the DevOps community should seek avenues for participating in or influencing further refreshes of these key reports. This may require membership in the appropriate professional organizations, building key relationships, and attending certain events.

Continuing education and training is based on a variety of commercial and non-profit frameworks:

- ITIL
- COBIT
- PMBOK
- TOGAF
- Scrum (Scrum Alliance)
- Kanban's various interpretations
- Scaled Agile Framework

---

[4] R. Shackelford, J. H. I. Cross, G. Davies, J. Impagliazzo, R. Kamali, R. LeBlanc, B. Lunt, A. McGettrick, R. Sloan, and H. Topi, *Computing Curricula 2005: The Overview Report* (ACM & IEEE, 2005) http://www.acm.org/education/curric_vols/CC2005-March06Final.pdf.

[5] The entire ongoing program can be accessed here: http://www.acm.org/education/curricula-recommendations.

Sometimes, these frameworks are introduced in traditional education (especially more vocationally-oriented programs), but more typically they are offered by commercial providers.

Additionally, vendors offer extensive training in their hardware and software products.

The Challenge of Vocationalism

One concern academics often raise in discussions with industry is whether suggested program changes are too "vocational"—that is, training that is very applied, without concern for underlying principles. However, DevOps does not imply the use of any specific languages or tools.

It is critical for DevOps, Agile, and digital advocates in discussions with academics to be able to discuss the theoretical and historical origins of our movement. There is a deep and rigorous intellectual history; for example, Scrum was invented based on Ken Schwaber's interactions with formal control theory and process engineering, and Jeff Sutherland's study of product development trends (e.g., Nonaka). Don Reinertsen also provides a comprehensive, principles-based explanation of modern product development that is influential in the Agile and DevOps movements.[6] Extensive, cited analysis of these intellectual foundations can be found in the Minnesota State report, *Renewing the IT Curriculum: Responding to Agile, DevOps, and Digital Transformation*.[7]

Infrastructure level concerns such as configuration management, infrastructure as code, virtualization, and serverless can be treated with academic rigor (e.g., exploring concepts of state management, declarative versus imperative approaches, Mark Burgess' work on Promise Theory).[8]

---

[6] D. G. Reinertsen, *The Principles of Product Development Flow: Second Generation Lean Product Development* (Redondo Beach, CA: Celeritas, 2009).

[7] Advance-IT Minnesota, *Renewing the IT Curriculum: Responding to Agile, DevOps, and Digital Transformation* (St. Paul, MN: 2016), https://www.researchgate.net/publication/309980701_Renewing_the_IT_Curriculum_Responding_to_Agile_DevOps_and_Digital_Transformation.

[8] M. Burgess, *Thinking in Promises*, (Sebastopol, CA: O'Reilly Media, 2015); M. Burgess, *Promise Theory: Principles and Applications (Volume 1)*, (CreateSpace, Independent Publishing Platform, February 5, 2014).

CONCLUSION

Organizations are looking for highly-skilled "T-shaped" DevOps talent to help them build better software faster and drive business success. Unfortunately, with the continuing acceleration of technology and the emergence of higher performing Agile/DevOps practices, our educational system is struggling to provide the relevant training to supply this talent. Because of this, employers are often required to implement their own internal DevOps training and career development programs. As those organizations develop career paths, skills assessments, and training programs, we suggest considering these three areas of guidance:

- **Early Career:** Technologists just entering the workforce should consider the more generalist nature of their role, focus on a growth mind-set, stay curious, and be optimistic. They should gravitate toward hands-on learning using a methodical scientific approach.
- **Middle Career:** Technologist in the middle of their career should make a concerted effort to stay "T-shaped" by keeping current on a broad range of technical skills and resisting the urge to become vertically specialized without horizontal, cross-disciplined collaboration.
- **Leader:** Technologists who have advanced in their career to become leaders should strive to be educators in practice, using their advanced skills and experience to teach, mentor, and develop others. They should exemplify cross-team collaboration, specifically leading efforts to build an internal and external community of technologists. These leaders should also bring a contagious "technical twinkle" to the organization through their enthusiasm, depth of knowledge, capability, and technical consistency.

The adoption of DevOps practices enables team members to bring their individual, specialized competences to bear across the entire problem domain. This changes the role of a team member from someone focused only on one specialty to someone having empathetic technical awareness of the challenges of the team as a whole. This enables the creation of better software, faster development, and can help drive business success.

It is our hope that the industry will champion the adoption of more DevOps concepts in the curriculum of our traditional educational providers. As a community, we have a role to play in advocating for better educational programs for the workforce.

APPENDIX: SKILLS FAMILIES

There are multiple models that organizations will use to promote their DevOps transformation,[9] but in every case, the intent is for these IT professionals to work collaboratively across cross-functional teams. These functional areas represent a collection of important skills families that should be developed across DevOps teams. A skills matrix or career ladder for DevOps professionals will include elements from each of these areas. Leaders can use these atomic units to create and expand teams. These can also be used to conduct assessments to fortify gaps or weaknesses.

- Management Skills
- Portfolio Management
- Product Management
- Scrum Master
- Business Analyst/Requirements
- Development and Engineering Skills
- User Experience
- Development (Product Creation)
- Coding (Software Engineering)
- Testing
- Data
- DevSecOps
- Architecture
- Quality
- Operational Excellence
- Service Desk/End User Support
- Production Support (e.g., SRE)
- Reliability Engineering
- Cloud/Platform
- Server Ops
- Network

[9] For more information on these models, see the 2016 Forum Guidance white paper *Thinking Environments*, http://itrevolution.com/devops_enterprise_forum_guidance.

- Monitoring
- Security Operations
- Audit and Risk Management
- Asset Management

Resources

Advance-IT Minnesota, *Renewing the IT Curriculum: Responding to Agile, DevOps, and Digital Transformation* (St. Paul, Minnesota, 2016). https://dynamicit.education.

C. T. Betz, "Influencing Higher Education to Create the Future DevOps Workforce," Slideshare.net, presented at the DevOps Enterprise Summit, 2017, posted by Charles Betz, January 6, 2017, https://www.slideshare.net/alphas0ng/higher-education-and-the-future-devops-workforce.

Association for Computing Machinery, "Curricula Recommendations," Association for Computing Machinery website, http://www.acm.org/education/curricula-recommendations.

Acknowledgments

Acknowledgments to Charles Betz, University of St. Thomas, for assistance in developing this paper.

Authors

- Jason Cox, Director, Systems Engineering, Disney
- Chivas Nambiar, Director, Systems Engineering, Verizon
- Paula Thrasher, Director Digital Services, CSRA