

How to Establish a Reskilling/Upskilling Talent Development Program for Software Engineering

By Bill Swanton, Distinguished VP Analyst
Deacon D.K Wan, Senior Director Analyst

How to Establish a Reskilling/Upskilling Talent Development Program for Software Engineering

Published 25 March 2021 - ID G00744643 - 13 min read

By Analysts [Bill Swanton](#), [Deacon D.K Wan](#)

Initiatives: [Software Engineering Strategies](#); [Applications and Software Engineering Leaders](#)

Software engineering leaders struggle to find and retain developers with the advanced skills needed for modern cloud architectures. This report shows how to build a formal skills development program to upskill or reskill existing employees and new hires to meet your evolving needs.

Overview

Key Findings

- Software engineering leaders struggle to hire developers in a competitive market who have the advanced technical skills required to work on new cloud architectures and digital initiatives.
- Software engineering leaders often focus on what employees did yesterday and not whether they are aligned with tomorrow's values and in-demand skills. This leads to an ever widening skills shortage, a disconnect in expectations and higher attrition rates.
- Software engineering leaders with growing backlogs of work and insufficient headcount often shortchange learning hours and budgets that focus on employees gaining new skills. This results in a stagnation of staff skills.

Recommendations

Software engineering leaders responsible for software engineering strategy and tasked with building talent capable of delivering successful digital initiatives in the cloud must:

- Plan to acquire skills required for digital success by creating a formal system of skills inventory and advancement.
- Motivate employees to actively broaden their skills and roles by transparently sharing the skills inventory and setting the expectation that all employees should be constantly upskilling into areas with the most need.
- Create learning opportunities by starting an Agile Learning program, including dedicating time to learning and creating a wide array of on-the-job learning activities (often employee-led) to help everyone develop.

Introduction

Software engineering leaders tell us that when they need new skills, they prefer to hire people who already have those skills. When it comes to modern cloud architectures and languages, those people are hard to find. Gartner's TalentNeuron data on job markets (see Note 1) shows that there are a high number of open positions for people with those skills, but relatively few candidates per position. Leaders often write into the job description a long list of these required skills, making candidates even harder to find. All these factors require organizations to offer higher salaries and, worse, delay critical work by months.

To overcome the skills shortage, software engineering leaders need to upskill and reskill their existing employees and new hires (entry level or with narrower skills). The idea is to develop each employee's skills *ahead* of demand, so that they can play a broader range of roles and contribute to technically challenging new efforts. Leaders also need to realize that the gap is continuing to widen as technologies mature and new ones are introduced. This is not a one-time project, but an ongoing program to become a continuous learning organization.

Learning and development must be a key focus of a software engineering leader's job, and a formal program is essential for success. This program can be run within the software engineering function or as part of a broader HR-run learning and development program.

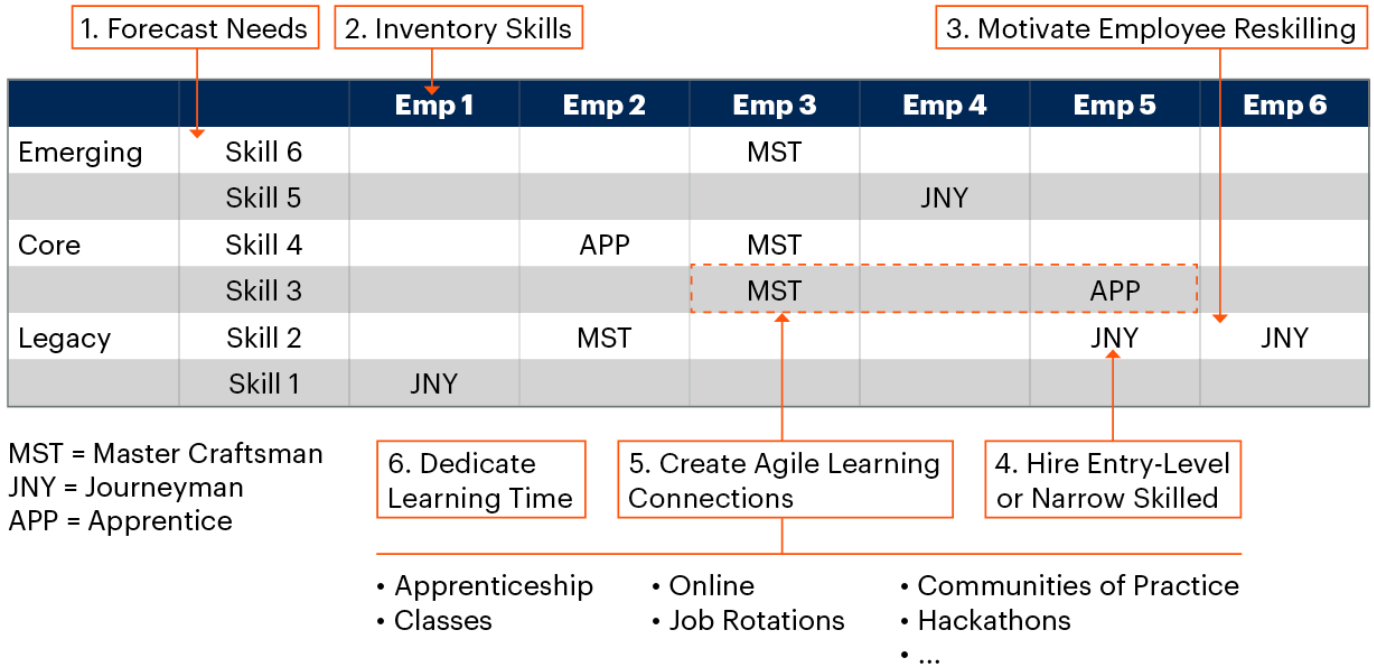
Gartner's research on Agile Learning shows that the payoffs of such a program include better attainment of business outcomes, upskilled/reskilled employees, the ability to adapt quickly to changing needs and an effective plan to meet skills needs. ¹

Figure 1 shows the six steps in creating a talent development program, which are each discussed in the analysis below. The program is built around a matrix of specific skills and competencies the organization requires, which have been classified based on trends of their present and future requirements. In addition, each employee is rated based on the level of proficiency in that skill.

The six steps are described in the Analysis section.

Figure 1. Training and Skills Development Program

Training and Skills Development Program



Source: Gartner
 743206_C

Analysis

Create a Formal System of Skills Inventory and Advancement

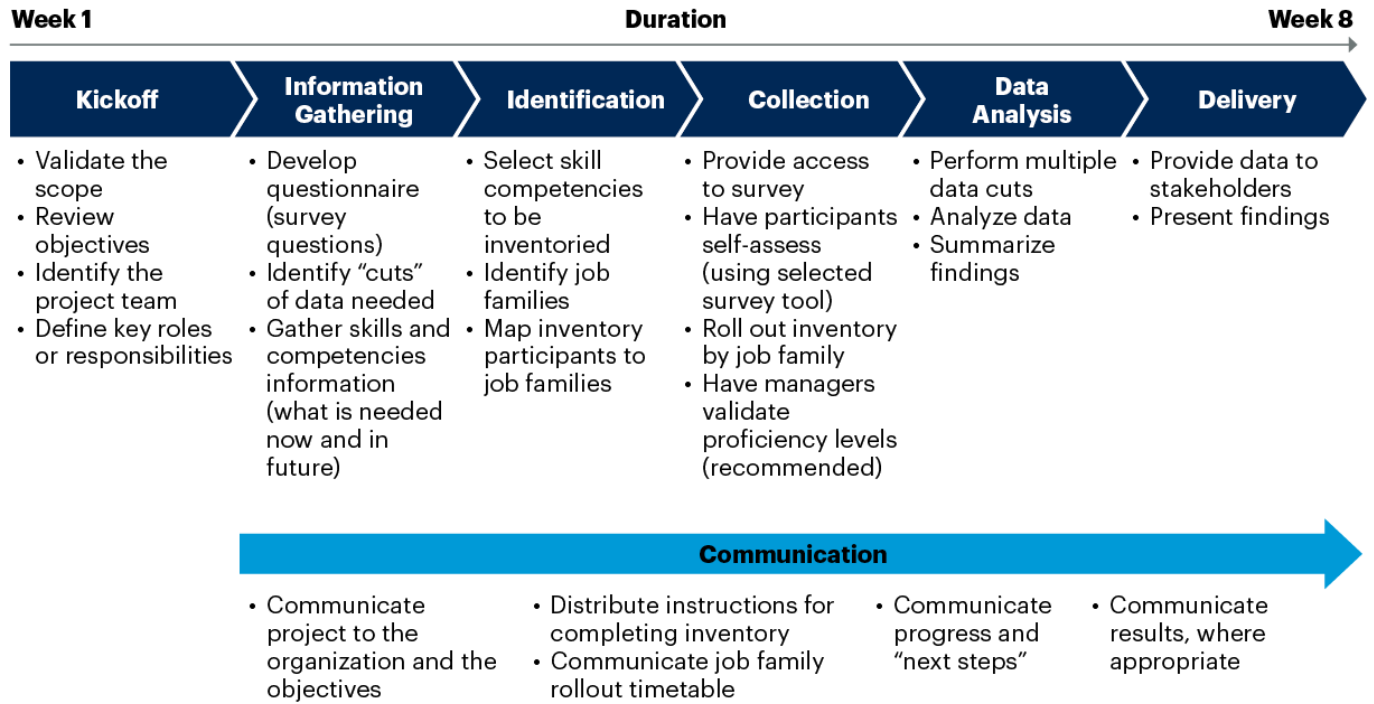
Companies vary widely in the formality of their learning and development approaches. While all have formal job roles and levels, specific skills are often tracked informally. In addition, many learning systems are primarily focused on general new hire onboarding, compliance and managerial skills training, leaving domain-specific training to functional leaders.

To close the skills gap, you should create a formal system to define and assess the skills and competencies needed now and in the future. As shown in Figure 2, this entails identifying the specific skills to be tracked, then surveying employees on their capabilities and vetting them with their managers. Communication of your purpose and process is critical to avoid creating fears and concerns among employees.

This process and Figure 2 are explained in more detail in [Assess Your IT Bench Strength for Digital Business Transformation](#).

Figure 2. IT Skills Inventory Process

IT Skills Inventory Process



Source: Gartner
744643_C

1. Identify the Skills and Competencies Critical to Your Future Success

It would be very difficult to enumerate every possible skill in the software engineering world. In creating your skills inventory, you need to identify the most important skills that determine whether a specific individual can do a particular kind of work, and those skills that you will need more of in the future.

You might use broader definitions for older skills that are declining in necessity, but be more specific for newer skills. You need to look about two years into the future and imagine what new systems or digital services you might be asked to build or what systems will need to be replaced, and what the likely architectures will be. You will need to work with business leaders and product managers to understand upcoming product needs, and with solution architects to understand the skills needed to build them.

You should be looking at a variety of things:

- **Competencies** – Business acumen, design thinking, etc., which you will want to develop in all employees.
- **Ways of working** – Agile, Kanban, DevOps, etc., especially if you are in the midst of a transformation.
- **Architectures and design** – SOA, MASA, cloud native, data architecture, user experience.
- **Technical skills** – Specific programming languages, development platforms, tools, etc.

The following Gartner research enumerates or gives examples of some of these skills:

- [Develop the Competencies Your Workforce Needs for the Digital Ecosystem](#)
- [Essential Skills for Modern Application Development](#)
- [Essential Skills for Application Architecture](#)
- [Essential Skills for Modern Integration Architecture](#)
- [Toolkit: Skills and Competency Assessment to Maximize Your IT Workforce Effectiveness](#)
- [How to Help Software Engineering Teams Modernize Their Application Development Skills.](#)
- [How to Help Software Engineering Teams Modernize Their Application Architecture Skills](#)
- [Prepare Now for the Future of Digital Product Management](#)

If your HR learning and development department has built skills matrices for other parts of the organization, get their help to build one for software engineering. Companies without such a department might set up a learning and development function within software engineering and, in large organizations, assign someone to lead the talent development program.

Now that you have enumerated the needed skills, you have to quantify the need. This should take the form of a forecast of how many people will need to have the skill in question to get the envisioned work done. This is not the same as forecasting the required headcount; ideally, employees will broaden their skills and become more versatile for staffing teams.

2. Inventory Current Employees' Skills and Level of Mastery

Next you need to inventory the skills your employees already have. This is done by surveying the employees on their skills and level of proficiency, and then having managers validate the results.

Before doing this, you need to communicate the purpose of the assessment to avoid creating fear and uncertainty about how the information will be used. Make it clear that you are trying to quantify where the organization has skills gaps and to help employees successfully gain new skills that are needed to make the employee and the organization successful.

You need a system to rate the proficiency of each employee in the skill. Two examples are:

- The ancient guild designations of apprentice, journeyman and master craftsman — see [Rebalanced Technical Skills Portfolio \(Nationwide\)](#).
- The five-level system (Being developed, Basic, Intermediate, Advanced, Expert) described in [Assess Your IT Bench Strength for Digital Business Transformation](#) and [Toolkit: Skills and Competency Assessment to Maximize Your IT Workforce Effectiveness](#).

Motivate Employees to Actively Broaden Their Skills and Roles

Part of becoming a learning organization is changing the culture so employees realize they are expected to broaden their skills on an ongoing basis. Both leaders and engineers have to change – leaders need to show they value learning on the job, while engineers cannot expect to be spoon-fed.

Make the results of your skills inventory (in aggregate) and forecast transparent to employees so they see where the gaps are. Make it clear what skills are:

- **Legacy/Niche** – Will be less needed in the future.
- **Core** – Are essential to running IT today.
- **Emerging** – Will soon be in greater demand.

3. Focus Employees on Taking Charge of Their Careers by Gaining Needed Skills

To successfully fill the skills gaps, software engineering leaders must motivate employees to broaden their skills. Start shifting performance discussions from “What have you done?” to “What have you learned?” Make it clear to employees that they need to drive their careers forward, and the best way to do that is to target learning new skills that are in short supply.

Emphasize that their value to the organization is having a wide range of current and emerging skills. This requires two-way transparency – communicating the company’s needs to employees, who in turn need to communicate their career aspirations. Use the intrinsic motivator of “mastery” (see Note 2) by recognizing new skills and proficiency levels and making sure the employee’s peers recognize them too. The people with those skills will have more opportunities to work on the most important and interesting products and applications. This aligns with the intrinsic motivator of “purpose.”

Keep in mind that development is not just for technical skills. It might also prepare the employee to take on a new role, such as moving from being a software engineer to a product manager (see [Create Career Lattices to Boost Talent Development and Drive Agile Transformation at Scale](#)).

4. Hire Entry-Level or Above Employees With Narrow Core or Emerging Skills and Upskill Them

Even as you get employees learning and gaining skills, you will still need new employees, and highly skilled employees are hard to find. You will need to find new people with narrow or lower-level skills and then grow them.

These may be recent graduates from college or people who have taken a high-intensity skill building class, such as a 12-week “coding camp.” Older workers who were, say, project managers or businessrelationship managers may take these courses to get current technical skills in order to be more employable. Look also within your own company, at people in adjacent roles, like technical or customer support – they know the products and customers and might be easily upskilled.

These hires should have a core skill that you need to build, and they should be striving to gain experience. Prioritize team-oriented competencies, such as collaboration and adaptability, as hiring criteria, and immediately start broadening new hires' skills and create opportunities to develop business acumen relevant to your industry.

Create an Agile Learning Program

Agile learning is a mindset and method of skills development, via iterative short bursts, applied in the flow of achieving outcomes that can dynamically adjust with changing needs. Agile learning is not just a training program that we can palm off on learning and development professionals; it is an attitude. It is a way of thinking. That is why it won't work unless we drive it into our culture, into employees' everyday habits, into the fabric of how our enterprise succeeds (see Figure 3 and [Agile Learning Manifesto](#)).

Figure 3. Agile Learning Manifesto

The Agile Learning Manifesto

Values

- ★ **Business outcomes**
over knowledge gained
- ★ **Growth mindset**
over current skill set
- ★ **Real-time embedded**
over training time offline
- ★ **Community compounding**
over individual practicing



Principles

- 1 Learning to earning
- 2 Motivation multiplier
- 3 Just-in-time microbursts
- 4 Dynamic pathways
- 5 Progressive layering
- 6 Flow of value delivery
- 7 Data-driven, AI-enabled
- 8 Socially amplified

Source: Gartner
730775_C

Gartner

5. Create Learning Connections for Employees to Learn New Skills on the Job

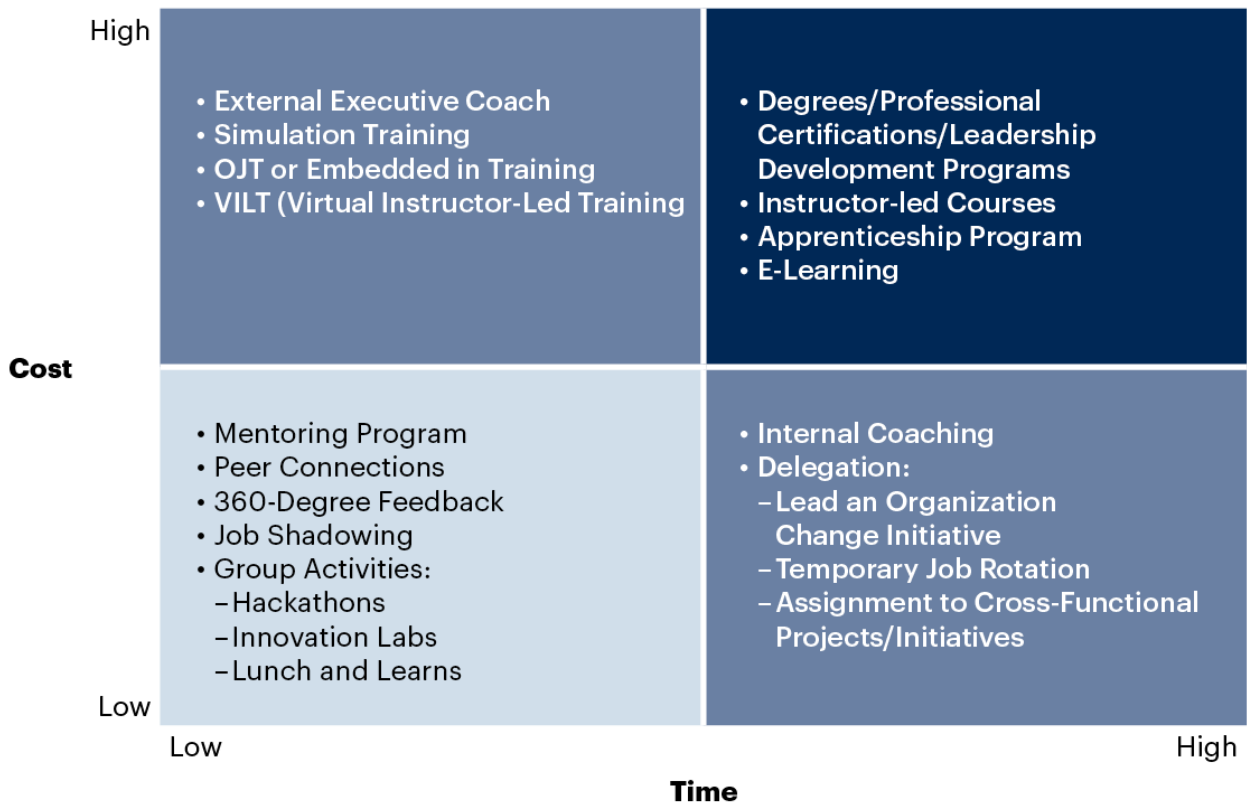
Software engineering leaders should create learning opportunities by connecting employees to interactions and work where they can learn the skills they have targeted. This means putting employees in situations where they can learn and, most importantly, apply new skills immediately. While there is a place for traditional and online courses, short learning moments on the job are low-cost, low time commitment and, as our Agile Learning research shows, highly effective.

Figure 4 shows different development approaches categorized by cost and time commitment. These are described in more detail in [Tackle the Talent Problem: Invest in Growing Your Own Employees](#). Morespecific advice for software engineering leaders can be found in [How to Help Software Engineering Teams Modernize Their Application Development Skills](#).

Apprenticeship is another approach, as discussed in the skills inventory section. More on this can be found in [Reskill Talent for Digital Business by Reimagining Apprenticeships and Getting Started on Your Apprenticeship Program](#).

Figure 4. Development Approaches

Development Approaches



Source: Gartner
744643_C

6. Dedicate Time to Learning Even Amid an Insurmountable Backlog of Work

You will never have enough people or time to deliver everything everyone wants, when they want it. You need to force stakeholders to prioritize their requests for each team, who can then work on them as quickly as the team’s capacity allows.

You must allocate time for learning to occur, or employees will spend all their time only using their current skills and not gaining new ones. Some learning activities can be integrated into the work, but may reduce productivity. For example, having an apprentice work on something under the supervision of a master will probably take longer and consume some of the master’s capacity.

You should also dedicate a fixed time period during the work week for everyone to participate in learning and cross-team activities. This could include master-taught classes, online individual learning, mentoring, proof-of-concept projects on new technologies and communities of practice. One organization we interviewed has dedicated one day a week to these activities, with teams working on their backlog four days a week. After three years, they have built up their skills base to the point that they were considering cutting that back to one day every two weeks.

Evidence

¹ [Survey Analysis: Agile Learning Delivers Superior Outcomes](#)

² D. Pink, "Drive: The Surprising Truth About What Motivates Us," Riverhead Books, 2011.

Acronym Key and Glossary Terms

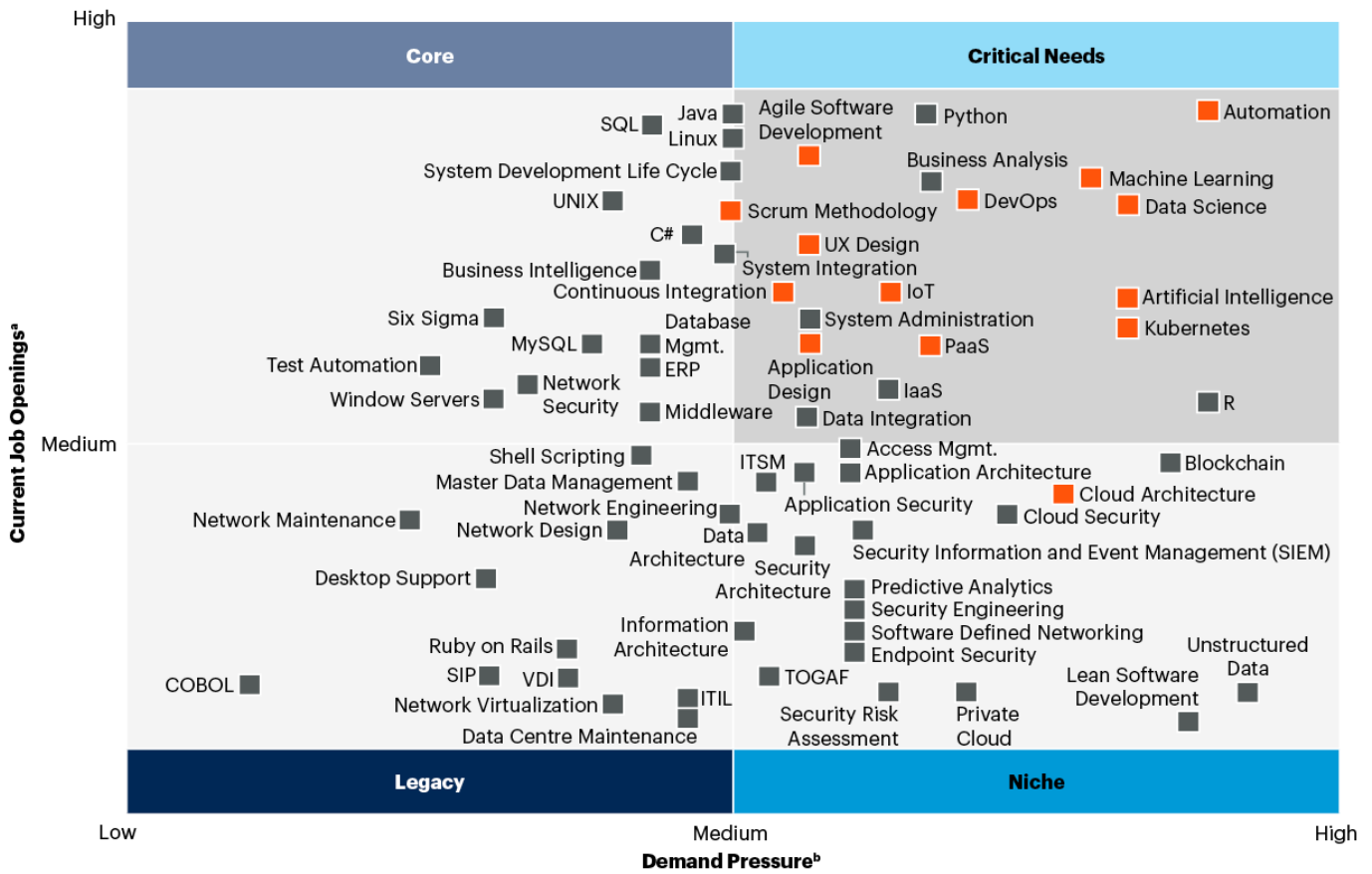
| | |
|------------|--|
| Competency | A set of characteristics of an individual that are observable, measurable and predictive of superior performance within a given role or job. They are how the job gets done. |
| Skill | The observable manual capabilities required in the execution of a job or task. They define what is performed within a job. |

Note 1: TalentNeuron Data

Using TalentNeuron, we sourced out data related to market wage, hiring scale and job postings related to the 68 most widely used and sought-after skills by IT executives. The infographic buckets these skills into four categories according to their viability and usability: legacy, core, niche and critical needs (see [2020 IT Skills Roadmap](#)).

Figure 5. IT Talent Quadrant, U.S., 2020

IT Talent Quadrant, U.S., 2020



Source: Gartner

Note: The tangerine dots represent the modern application development skills.

^aThe current number of job postings per skill recorded at the end of October 2019.

^bMeasure of the number of candidates available for a given job opening. High demand pressure means a small number of candidates per job opening.

744643_C

Note 2: Motivation

As a leader, motivating employees to do the right things is the core of your job. The passion we strive to instill in employees is best expressed in the terms Daniel Pink uses in “Drive: The Surprising Truth About What Motivates Us.”² Software engineering leaders must understand employee development in the following terms:

- **Autonomy** – As a leader, get out of the way and let the teams figure out the best way to get the job done. Help as needed to remove roadblocks and other impediments to their progress.
- **Mastery** – Help individuals develop skills and roles. Ensure that they receive recognition when they master a skill – not just from managers, but also from their peers.
- **Purpose** – People need to feel that their work is important, and that their work is helping others to thrive. You need to connect them with this purpose.

With flatter agile organizations and self-organizing work teams, career advancement for most employees will not involve moving up conventional career ladders and managing people. Learning and recognition of skills mastery in diverse technology domains become powerful motivators for engineers, helping to improve morale and retain talent (see [Changing Leadership and Motivation in a Product-Centric Development Organization](#) and [How CIOs Can AMPLify Employee Engagement Through Autonomy, Mastery and Purpose](#)).

Recommended by the Authors

[Renew Retention Strategies to Retain Technology Talent for Digital Business Success](#)

[Using a Digital Talent Management Framework to Build a Digital-Ready Workforce](#)

[Assess Your IT Bench Strength for Digital Business Transformation](#)

[Agile Learning Manifesto](#)

[Tackle the Talent Problem: Invest in Growing Your Own Employees](#)

[Rebalanced Technical Skills Portfolio \(Nationwide\)](#)

[Toolkit: Skills and Competency Assessment to Maximize Your IT Workforce Effectiveness](#)

[Changing Leadership and Motivation in a Product-Centric Development Organization](#)

[Case Study Detail: Customized Delivery Capability Development for Product Teams to Achieve Business Outcomes \(Intrado\)](#)

[Ignition Guide to Creating a Competency Development Plan for Agile Teams](#)

Recommended For You

[How to Help Software Engineering Teams Modernize Their Application Development Skills](#)

[Innovation Insight for Chaos Engineering](#)

[Reduce Friction to Boost Software Engineering Team Productivity](#)

[How to Help Software Engineering Teams Modernize Their Application Architecture Skills](#)

[How to Define and Guide Citizen Development Practices](#)

© 2021 Gartner, Inc. and/or its affiliates. All rights reserved. Gartner is a registered trademark of Gartner, Inc. and its affiliates. This publication may not be reproduced or distributed in any form without Gartner's prior written permission. It consists of the opinions of Gartner's research organization, which should not be construed as statements of fact. While the information contained in this publication has been obtained from sources believed to be reliable, Gartner disclaims all warranties as to the accuracy, completeness or adequacy of such information. Although Gartner research may address legal and financial issues, Gartner does not provide legal or investment advice and its research should not be construed or used as such. Your access and use of this publication are governed by [Gartner's Usage Policy](#). Gartner prides itself on its reputation for independence and objectivity. Its research is produced independently by its research organization without input or influence from any third party. For further information, see "[Guiding Principles on Independence and Objectivity](#)."

Become a Client

Get access to this level of insight all year long — plus contextualized support for your strategic priorities — by becoming a client.

gartner.com/en/become-a-client

U.S.: 1 800 213 4848

International: +44 (0) 3331 306 809

About Gartner

Gartner is the world's leading research and advisory company and a member of the S&P 500. We equip business leaders with indispensable insights, advice and tools to achieve their mission-critical priorities today and build the successful organizations of tomorrow.

Our unmatched combination of expert-led, practitioner-sourced and data-driven research steers clients toward the right decisions on the issues that matter most. We are a trusted advisor and an objective resource for more than 14,000 enterprises in more than 100 countries — across all major functions, in every industry and enterprise size.

To learn more about how we help decision makers fuel the future of business, visit gartner.com.