



*An Online PDH Course
brought to you by
CEDengineering.com*

Transferring Knowledge To The Next Generation Of Engineers

Course No: K05-010
Credit: 5 PDH

Richard Grimes, MPA, CPT



Continuing Education and Development, Inc.

P: (877) 322-5800
info@cedengineering.com

Table of Contents

Introduction	1
First Things First.....	4
More About the “Who”	8
A Snapshot of the Generations	8
Baby Boomers	9
Generation X.....	10
Millennials	11
Generation Z	11
Some Direct Generational Comparisons	12
Why Knowledge Transfer (KT) Matters Today.....	14
Who Wishes They Had Done This?	16
Key Contrasts and Takeaways	18
Coaching vs. Mentoring.....	19
What’s In It For Me?.....	20
Selecting Participants.....	23
Example 1: Self-Assessment Questionnaire for Junior Engineers	23
Example 1: Self-Assessment Questionnaire for Senior Engineers.....	24
Example 2: Quick Matching & Compatibility Tool	25
Example 3: An In-depth Informational Gathering Tool.....	26
Example 4: Matching Methods with Knowledge	28
Reverse KT Strategies	32
Why Reverse KT Matters in Engineering.....	32
Key Benefits for Engineering Organizations.....	32
Practical Reverse KT Strategies	33
Generational Challenges and How to Overcome Them	34
Implementation Tips for Engineering Firms.....	34
Potential Engineering KT Scenarios.....	35
AI Tool Adoption	35
Sustainability Perspectives	35
Communication Style	36

Measuring Success and ROI.....	37
Productivity Gains	38
Quality Improvements / Defect Reduction	38
Knowledge Retention & Reduced Ramp-up Time	38
Innovation / Idea Generation	39
Retention & Engagement.....	39
Measuring Perceptions.....	39
Likert Survey Design Tips	41
Overcoming Resistance	44

Introduction

Looking back over my years as Manager of Employee Training and Development for a large engineering and construction company, I feel some regret. More than 30 years later, I now clearly see how much precious experience, knowledge, and human potential we let slip away.

We had bright, eager young engineers ready to grow, and seasoned seniors carrying decades of hard-earned wisdom — yet we had no meaningful system in place to transfer that knowledge. We missed the chance to truly develop our new talent while honoring and preserving the legacy of those who were retiring. That missed opportunity still weighs on me today which led to the creation of this course.

If there were a way I could go back with today's wisdom and the energy I had then, this would be my way of bridging that gap of our senior staff's hard-earned savvy and field smarts with the unlimited potential of our newly-hired EITs and freshly-minted PEs.

I also remember that engineers always ask, “so what?” when presented with non-technical, “soft skills” topics, so I'll lay it out as I would if I were pitching this idea to our engineering execs and senior staff.

I would say,

“I think the newer engineer benefits would include:

- a. **Faster Mastery of Real-World Skills:** *Instead of learning through trial and error (often an expensive and frustrating process), new engineers would receive direct, practical guidance from seasoned professionals who have already solved complex problems in the field.*
- b. **Preservation of Institutional Knowledge:** *They would absorb the "unwritten rules," lessons from past failures, and nuanced judgment that textbooks and formal training rarely capture. This shortens the typical 5–10 year ramp-up period to full productivity.*
- c. **Increased Confidence and Reduced Risk:** *Having a trusted mentor provides a safety net for asking "dumb" questions, reviewing designs, and making decisions. This leads to fewer costly mistakes and faster professional confidence.*
- d. **Career Development and Belonging:** *Structured mentoring helps new engineers understand career paths, develop leadership skills, and feel genuinely valued. This dramatically improves engagement, job satisfaction, and long-term retention.*
- e. **Broader Perspective:** *Exposure to the wisdom of senior engineers helps them develop better judgment, foresight, and engineering intuition.*

Of course, the senior participants gain something, too:

- **Legacy and Sense of Purpose:** Many retiring engineers feel a quiet sadness watching their hard-earned knowledge walk out the door with them. Mentoring allows them to leave a lasting legacy — knowing their experience will live on through the next generation. This often brings profound satisfaction and emotional closure.
- **Respect and Recognition:** Being formally asked to mentor signals that their career contributions are still highly valued. It honors their expertise instead of making them feel obsolete as retirement approaches.
- **Intellectual Stimulation:** Teaching forces seniors to organize and articulate their knowledge, often leading to new insights about their own work. Many discover they learn as much as they teach.
- **Reverse Mentoring Opportunities:** In a healthy program, older engineers are also exposed to new technologies, fresh approaches, and digital tools from their mentees. This keeps them engaged and relevant.
- **Smoother Life Transition:** Mentoring provides a meaningful way to gradually reduce workload while staying connected to the profession. It softens the identity loss many seniors experience when they fully retire.
- **Personal Fulfillment:** Numerous studies and personal accounts show that mentoring is one of the most rewarding activities for experienced professionals. It gives them a renewed sense of contribution and impact.”

I think that if I could have presented that list of benefits to our executive and senior leadership, they would have insisted we implement the program RIGHT NOW!!

Unfortunately, I can't travel back through time to preserve some of that hard-earned wisdom we lost and retain some of the rising stars who left us out of career-smothering frustration but IT'S NOT TOO LATE FOR YOU!



I've written this as a "How To" guide for any leader who finds themselves in a situation similar to mine so long ago. It's designed to put into practical terms and methods an engineering talent development and

retention program while transferring critical, hard-earned knowledge and wisdom that will prove invaluable to your company and its future.

I hope it helps you preserve your young high potentials and honor that senior talent – both of which are key to your future!

Richard Grimes, 2026

First Things First

First things first. We will assume that you have made a similar pitch to your executives they said GET STARTED NOW!

If you have never attempted an experiment like this going into uncharted human relations territory (and you don't have the luxury of a strong Human Resources group to take the lead), consider my experience of many years doing projects like this.



I have found there is a similar thread running through them: *it usually takes three iterations before you're comfortable with the process.* ("Comfortable" means it's running smoothly enough that you can relax but there may still be opportunities for fine tuning here and there.)

Your first attempt will be just enough to get it running so you can see whether it will actually work. (Imagine a caveman fashioning two wheels out of stone that is reasonably round with a stick going through the center as a makeshift axle with a basket above creating a load-carrying device.)

Your second attempt may involve replacing the participants (like rounding the wheels so it rolls better) and your third attempt may have the right people in the right places at the right time (which is like putting wheels at each corner with 2 axles creating a cart that's easier to pull than a one-axle carrier that you have to balance and push.)



Once you reach the cart stage, step 3, in your experiment, you're "comfortable" with the outcome unless a really great idea comes along (like installing an engine and inventing a convertible) and you refine it even further.

I will also assume that you have an engineering background (instead of Human Resources or other soft-skills) and think in an orderly, linear, and sequential fashion. If that's true, then we will look at this from the **Who, What, Where, When, and How** perspective of this opportunity.



WHO: Since this is an experimental program with many people (and a few skeptics) watching to see if it works (or you to fail), you will want to give it every chance for success.

Therefore, select the most mature, likable, and talented “newbie” you have and the friendliest, warmest, and wisest “senior” available to be your foundational participants. This choice of your first pair is a critical decision not to be hurried and DO NOT just select anyone available or not doing anything else.

They will have two critical jobs:

Another Benefit

The senior member of the pair could be your manager of the program going forward since his/her workload is probably diminishing. Who would have a more intimate, thorough knowledge and appreciation of the program than someone who helped birth it?

1. Engage in a genuine teacher-student relationship which flips roles frequently as the student becomes the teacher and becomes the student again and,
2. As co- program developers with you in this new organizational experiment. **MAKE SURE YOU HAVE A SENIOR WHOSE EGO WILL ALLOW THEM TO BECOME A STUDENT AT TIMES AND A JUNIOR WHO IS NOT SMUG AND ARROGANT ABOUT THEIR TECHNICAL SKILLS.**

There are several questionnaire samples beginning on page 22 that will help you select your first participants.

WHAT: Explain to them the program – even giving them copies of my introduction a few pages back to help them understand fully what you’re attempting to do here. Make sure they understand they are part of an important experiment that has the potential to help many people in your organization, both now and in the future, and they are pioneers in this grand experience. (This plays to their egos.)

Tell them about the three iteration examples above so they realize you do not have all the answers for the final product and that you will be counting on them to furnish many suggestions for refinement and improvement since it's an experimental program. Help them understand how vital they are to the success of this program. That sometimes one will be the teacher of a familiar topic while the other one learns and the next day the roles may be reversed.

That is why it is so critical that you get the right pair upon which to build a successful foundation. (There's nothing stronger than a first impression. You want them to make the best first impression possible for your program.)

WHERE: The “**where**” of the program is almost as important as the “**who**” for the first few iterations until it gets running smoothly and you have a few proud examples to brag about. (This makes me think about publicity which we'll discuss later.)

The essential element of the “where” is that the initial participants have easy access to each other as much as possible. Whether in the home office or on a remote project site, the first two should be at the same location.

There will be many opportunities for conversations together at lunch, in one's office, riding to a meeting, stopping by one's desk for a demonstration of something. Proximity strengthens the relationship and they should be in the same location – at least until a strong bond is formed – if possible.

“Soft Opening?”

Many new restaurants have “soft openings” before they fully announce they are open and ready for customers.

This gives them a chance to practice their menu, observe staff in actual conditions, and just make sure everything works like it should before “going public.”

Maybe you and your chosen pair work on it unannounced for a while before “going public” to the whole organization.

This would give you two credible witnesses to the value of the program.

Later, as your program matures, you can gradually experiment with remote pairing using Zoom, MS Teams, or other similar platforms to facilitate their exposure to each other after a face-to-face initial meeting (if possible with scheduling and expense considerations.)

WHEN: This is your decision because it is built around your schedule. But I strongly suggest that you have at least 90 days available that you can dedicate to this. **(90 days to get it going – NOT 90 days for the program to run. You hope it runs for years and becomes embedded in the corporate culture.)**

So it will probably not be a full-time effort but there will be a lot of prep on your part studying this manual and becoming familiar with the characteristics of the two or three generations with which you'll be working.



You will want to make sure you have prepared a few public relations announcements to recruit participants.

Will you need time to sell it to some of the senior “old school” leaders who may not be supportive of these “new kids getting chances they never had!”?

Just like with your design-build projects, there is rarely a perfect launch – mostly it’s “let’s get started and adjust as we go along until everything is clicking.”

HOW The “How” is the reason for the rest of this course and is explained in the following pages.

More About the “Who”



Before we get too far into this subject, let’s take a moment to understand something about the focus of this course – the people.



As of this writing in 2026, the traditional age of retirement in the US is 65 years old meaning anyone still working will have their birthdays in the 1960s and their ages also in their 60s. These are the **Baby Boomers** (born 1946-1964) and are actively planning on retirement if they have not already left.

Next are the **Generation Xers** born between 1965-1980 making them in the 46-61 age range. Probably not retiring anytime soon but it is a growing prospect on their horizon – especially if health issues are present.

Then the **Millennials** come next (born 1981-1996) with an age range of 30-45 followed by **Generation Z**, (1997-2012), your youngest and most tech-savvy group aged 29 and below.

A Snapshot of the Generations

Here’s a balanced overview of pros, cons, and obstacles for each major generational group in a knowledge transfer (KT) program that may be useful for you.

Typical engineering environments often involve deep technical expertise, problem-solving under pressure, evolving tools/tech, regulatory/compliance knowledge, and project-based work—factors that amplify generational dynamics.

Baby Boomers

(Born 1946–1964) They are typically the most experienced group, often in senior or advisory roles. Many are approaching retirement or have decades of institutional knowledge. (Since I’m a Boomer, I have first-hand knowledge about them and will offer a little more about them since they may seem difficult with whom to work.)

- They prefer formal communication, so managers should use structured and formal methods to communicate with them. (Texting with them instead of calling or talking may not be as effective as with the other generations. Also, the concept of “casual Friday” or the increasing trend of casual office wear may seem “unprofessional”.)
- They are competitive and workaholics, so they might need clear goals and recognition for their achievements.
- They value job security, so they might be less likely to take risks that could threaten their position.
- Early on, they had “pension plans” into which a portion of their salary would go every pay period for retirement. The longer they stayed with the company, the larger their pension amount. If they left, the money stayed with the employer – few ever left voluntarily.



Pros: They can bring deep domain expertise, historical context, lessons from past failures/successes, organizational politics navigation, and long-term strategic thinking.

- Strong mentoring mindset; many enjoy legacy-building through teaching.
- Reliability, work ethic, and patience in complex, high-stakes engineering projects.
- Reverse mentoring opportunities allow them to stay relevant (e.g., learning new tools).

Cons/Obstacles:

- Potential resistance to rapid tech changes or “new” methodologies (e.g., agile vs. waterfall, AI tools).
- Risk of knowledge hoarding or frustration with perceived lack of junior commitment. (*If I teach someone my job, why would they keep me?*)
- Time constraints or skepticism about formal programs (*“I learned on the job. No one taught me how to do it.”*).
- Physical/energy limitations or discomfort with digital collaboration tools.
- Generational stereotypes (e.g., seen as “out of touch”) can hinder open dialogue.

Key Program Obstacle: Impending retirements create urgency for knowledge capture before loss of tacit knowledge.

Generation X

(Born 1965–1980) Often in mid-to-senior leadership or technical expert roles. They are a “bridge” generation with experience from both analog and digital eras.



Pros: Highly independent, adaptable, and pragmatic; strong problem-solvers who value results over process.

- Good balance of technical depth and practical experience; effective at translating between senior and junior perspectives.
- Skeptical but open-minded—can challenge outdated practices while respecting fundamentals.
- Reliable mentors who appreciate work-life balance and can model sustainable careers.

Cons/Obstacles: Cynicism or “been there, done that” attitude may reduce enthusiasm for programs.

- Preference for independence can lead to lower engagement in structured mentoring.
- Caught in the middle: managing upward (Boomers) and downward (younger gens) expectations.
- Potential burnout from carrying institutional load during Boomer retirements.

Key Program Obstacle: They may feel overlooked in favor of Boomers (for wisdom) or Millennials/Gen Z (for innovation).

Millennials

(Born 1981–1996) Now a large part of the engineering workforce, often in mid-level technical and leadership roles.

Pros: Collaborative, achievement-oriented, and comfortable with both traditional and modern methods.

- Strong at integrating tech with processes; good at documentation, knowledge sharing via digital tools, and cross-functional work.
- Value feedback, growth, and purpose—motivated to participate in development programs.
- Bridge builders: understand Boomer/Gen X contexts while relating to Gen Z.

Cons/Obstacles: Stereotypes of being “entitled” or job-hoppers can erode trust from older generations.

- May lack depth in certain legacy systems or long-term project perspectives.
- Desire for rapid feedback and growth can clash with slower-paced traditional mentoring.
- Work-life balance priorities might limit extra time commitment.

Key Program Obstacle: Balancing their own career progression needs with mentoring responsibilities.

Generation Z

(Born 1997–2012) Entering or early in careers; the most digitally native group, growing rapidly in engineering roles.

Pros: Tech-fluent (AI, digital tools, data analytics, simulation software); bring fresh ideas and innovation.

- Value diversity, sustainability, and continuous learning—align well with modern engineering challenges (e.g., green tech, ethics).
- Comfortable with reverse mentoring (teaching seniors digital skills).
- Seek purpose and feedback; can energize programs with enthusiasm and new perspectives.

Cons/Obstacles: Perceived lack of patience, resilience, or deep foundational knowledge (“Google everything” mindset).

- Preference for quick, bite-sized learning and frequent feedback vs. long apprenticeships.
- Communication style differences (e.g., preference for async/digital over face-to-face).
- Higher anxiety or need for psychological safety; may struggle with ambiguity in complex engineering problems.

Key Program Obstacle: Building credibility with older generations while gaining enough hands-on experience.

Common Cross-Generational Obstacles in Engineering KT Programs:

- Differing communication and learning styles (formal vs. informal, in-person vs. digital).
- Stereotypes and biases that reduce openness.
- Time pressures in project-driven environments.
- Lack of structured program design (training for mentors/mentees, clear goals, metrics).
- Knowledge hoarding vs. over-reliance on quick searches.

Some Direct Generational Comparisons

Communication Preferences:

“Science is about knowing, engineering is about doing.” — Henry Petroski. Reminds participants that knowledge transfer must lead to practical application and real-world problem-solving.

- Baby Boomers: Formal, structured
- Generation X: Casual, flexible
- Millennials: Digital, collaborative
- Generation Z: Digital, quick, informal

Work Style:

- Baby Boomers: Competitive, structured
- Generation X: Independent, flexible
- Millennials: Team-oriented, purpose-driven
- Generation Z: Entrepreneurial, adaptable

Motivation Factors:

- Baby Boomers: Job security, recognition
- Generation X: Work-life balance, autonomy
- Millennials: Meaningful work, feedback
- Generation Z: Diversity, flexibility, entrepreneurship

Expectations from Managers:

- Baby Boomers: Opportunities for advancement, recognition
- Generation X: Trust, flexibility

- Millennials: Mentorship, regular feedback
- Generation Z: Support for innovation, inclusive environment

Why Knowledge Transfer (KT) Matters Today

If you are considering the establishment of a program like this, you are looking at a sizable commitment of time and challenging the organizational culture – doing something never attempted before and going contrary to embedded groupthink of “if it isn’t broken, don’t fix it!”

We will spend some time making sure you understand – and can explain when the opportunity arises – why we should spend time fixing something that actually IS BROKEN but *we may not realize it*.

An aging workforce refers to a rising average employee age, driven by longer lifespans, delayed retirements, and demographic shifts (e.g., Baby Boomers born 1946–1964 now retiring en masse, combined with lower birth rates in many developed countries).

This creates a knowledge retention crisis—the *loss of critical institutional, technical, and experiential knowledge as veterans exit*—leading to multiple interconnected problems. Let’s explore the primary concerns.

1. Massive Knowledge Loss (The Core Crisis)

- **Tacit knowledge**—the intuitive, experience-based know-how that’s hard to articulate (e.g., a mechanic’s “feel” for when a machine is about to fail, a negotiator’s read on subtle cues, or an engineer’s pattern recognition from years of troubleshooting)—is central to the knowledge retention crisis in aging workforces. Unlike explicit knowledge (documents, manuals, data), it resides in people’s heads and bodies and is best transferred through interaction, observation, and practice. Capturing it involves externalization (making it more shareable) rather than full codification
- **Retirement wave:** In sectors like manufacturing, engineering, nuclear power, utilities, aerospace, healthcare, and government, large cohorts are retiring simultaneously. When they leave, organizations lose “corporate memory,” leading to repeated mistakes, longer problem-solving times, and reinventing solutions.
- **Result:** Reduced organizational capability, safety risks (e.g., in process industries or aviation), and slower innovation transfer to younger staff.

2. Succession Planning and Talent Pipeline Failures

- Many companies failed to invest in robust mentorship, apprenticeships, or knowledge-capture programs during boom years.
- Younger generations (Millennials, Gen Z) are smaller in key regions and often pursue different careers (tech, services over traditional trades).

- Skills mismatch: Retirees hold deep domain expertise; new entrants bring digital fluency but lack experience. **Bridging this gap takes years.**
- Outcome: Prolonged vacancies, over-reliance on expensive contractors/consultants, or promotion of underprepared staff.

3. Productivity, Innovation, and Adaptability Challenges

- **Short-term:** Experienced older workers often deliver high productivity, reliability, and quality. Their exit can cause immediate dips.
- **Longer-term:** Aging teams may become more risk-averse, slowing adoption of new technologies (AI, automation, agile methods). Cognitive declines in some individuals (though highly variable) can affect complex decision-making.
- **Intergenerational friction:** Differences in work styles, communication (e.g., digital vs. in-person), and expectations can reduce collaboration if not managed. (We have some ideas later to help overcome this obstacle.)
- **Innovation stagnation:** Loss of "grey hair" knowledge transfer can weaken R&D or problem-solving depth in knowledge-intensive fields.

4. Economic and Operational Costs

- **Higher labor costs:** older workers command higher salaries, pensions, and benefits. Healthcare expenses rise significantly (chronic conditions, longer recovery from injuries).
- **Absenteeism and presenteeism:** More medical leave or reduced capacity while at work.
- **Recruitment and training burdens:** Intense competition for scarce young talent drives up wages. Onboarding and training costs soar as knowledge must be rebuilt.
- **Macro impacts:** Labor shortages contribute to inflation, slower GDP growth, and strained social security/pension systems. Countries like Japan, Germany, Italy, and parts of the US/Europe face this acutely.

5. Sector-Specific Vulnerabilities

- **Critical infrastructure:** Energy, water, transportation—retirements threaten reliability and safety.
- **Healthcare:** Aging nurses/doctors exacerbate shortages while patient populations also age.
- **Public sector:** Government agencies and defense often have the oldest workforces, risking service delivery failures.
- **STEM fields:** Engineering and trades show acute "brain drain" in Western economies.

6. Broader Societal and Strategic Risks

- **Competitiveness:** Nations/companies with poor knowledge retention lose ground to those with better demographics or management (e.g., via immigration, automation, or strong training cultures).
- **Inequality:** Certain regions or demographics (rural areas, specific ethnic groups) suffer more.
- **Black swan amplification:** Crises (pandemics, supply chain shocks, wars) hit harder when experienced personnel are gone.

In summary, an aging workforce isn't inherently bad (experience is valuable), but the rapid, uneven exit without adequate knowledge transfer mechanisms creates a compounding crisis: **lost productivity, elevated costs, heightened risks, and weakened resilience.**

It reflects deeper demographic realities that societies must address through policy, education, and workplace innovation.

Who Wishes They Had Done This?

Here are three real-world examples of this process if you need ammunition for your presentation to your executives.



These are engineering case examples contrasting successful and failed (or challenged) intergenerational knowledge transfers. These draw from aerospace, infrastructure, and high-stakes industrial contexts, highlighting the role of structured approaches versus gaps in planning, culture, or execution.

1. Boeing (Aerospace Manufacturing) – Mixed but Often Cited as Failure/Challenged

- **Context:** Boeing faced a significant aging workforce and retirement wave. As experienced engineers and mechanics retired, institutional "tribal knowledge" about assembly processes, tooling, and problem-solving was lost.
- **Failed/Challenged Aspects:** During 737 production ramp-ups, the company reportedly had to rehire hundreds of retirees to troubleshoot issues on the assembly line. Critical undocumented know-how on manufacturing nuances led to delays, quality problems, and increased costs.

Broader critiques (e.g., around the 737 MAX program) also point to failures in transferring design rationale, risk knowledge, and lessons between teams/generations, exacerbated by cost-cutting, outsourcing, and cultural shifts that reduced emphasis on deep engineering experience sharing.

- **Lessons:** Without proactive, embedded transfer (e.g., mentoring tied to daily work), rapid retirements create immediate operational pain. Boeing has invested in knowledge management, but execution gaps illustrate the high cost of loss in complex, safety-critical systems.



2. NASA Jet Propulsion Laboratory (JPL) – Successful Structured Transfer

- **Context:** JPL, a project-based organization managing Mars missions and other explorations, deals with long project cycles and an aging cadre of senior engineers holding vast experiential knowledge on systems engineering, mission design, and troubleshooting in extreme environments.
- **Success Factors:** JPL implemented deliberate knowledge management under a *Chief Knowledge Officer*, including:
 - Hands-on training programs (e.g., building miniature Mars rovers for juniors to gain practice-based experience).
 - Lessons-learned databases, communities of practice, expertise locators, and balanced documentation with relational transfer.
 - Efforts to instill senior "practice-based experience" in juniors while reusing designs innovatively across missions (e.g., Mars 2020).
 - **Outcomes:** Better continuity between projects, reduced reinvention, and accelerated development for new engineers. This proactive, multi-method approach (mentoring, simulation, repositories) mitigates the retirement cliff in a high-complexity environment.

3. **Nuclear Power Industry** (e.g., IAEA-Supported Programs & Utilities) – Successful Proactive Transfer vs. General Risk of Failure

- **Context:** Nuclear plants face one of the most acute aging workforce issues due to specialized, safety-critical tacit knowledge (reactor operations, maintenance, decommissioning) that is heavily regulated and hard to replace.
- **Successful Examples:** Many utilities and agencies (supported by IAEA guidelines) use structured programs: mentoring/apprenticeships, job shadowing, incumbent interviews, storytelling, simulations, phased retirements, and digital capture tools. For instance, knowledge retention initiatives in operating organizations have preserved expertise for safe operations and new builds by pairing veterans with juniors years in advance and creating validated repositories.
- **Contrast with Failure Risks:** Where programs are absent or inadequate, plants experience knowledge gaps leading to safety incidents, extended outages, higher training costs, or licensing issues. Studies highlight "lack of knowledge acquisition" impairing reviews and operations when tacit insights aren't transferred.
- **Outcomes:** Proactive nuclear programs demonstrate measurable improvements in workforce readiness and safety culture, serving as a model for other high-stakes engineering fields.

Key Contrasts and Takeaways

- **Success Patterns:** Long-term planning (start 2–5+ years early), blend of human methods (mentoring, shadowing) with tech (videos, AI-assisted capture), leadership buy-in, and integration into daily work/culture. NASA JPL and nuclear examples emphasize relational + codified approaches for tacit knowledge.
- **Failure Patterns:** Reactive approaches, over-reliance on documentation alone, cost pressures reducing headcount/mentoring time, or cultural barriers (e.g., seniors not sharing, juniors not asking). Boeing illustrates how this amplifies in scaled manufacturing or complex systems.
- **Common Engineering Impact:** Failures raise safety risks, costs, and delays; successes preserve innovation capacity and competitiveness amid demographic shifts.

Coaching vs. Mentoring



Mentoring and coaching are both vital for talent development in engineering environments, but they differ significantly in approach, duration, focus, and application.

They are often confused or used interchangeably yet distinguishing them helps organizations address the knowledge retention crisis in aging workforces more effectively.

Here are the essential differences.

(The desired outcomes are identical, though.)

*We're not calling this a "mentoring program" because that can lead to many long and unproductive discussions of exactly what it means. However, since we're talking specifically about KNOWLEDGE TRANSFER, that is what we will call it here and avoid all the non-productive arguments..

Dimension	Mentoring*	Coaching
Duration	Long-term (months to years, sometimes ongoing)	Short-term (weeks to months, with clear endpoints)
Focus	Holistic career & personal development; tacit knowledge transfer	Specific skills, performance gaps, or immediate goals
Approach	Advisor/role model: Shares personal experiences, advice, networks	Facilitator: Asks powerful questions to help the person find their own solutions
Direction	Often student -driven (agenda set by student's needs) NOTE: Sometimes the "student" will be the senior partner learning a new technical skill.	Can be coach- or organization-driven (targeted outcomes)
Expertise Needed	Deep domain/industry experience (e.g., senior engineer)	Coaching skills/training (not always same technical background)
Relationship	Relational, trust-based, often informal	Structured, goal-oriented, more formal
Outcomes	Career growth, confidence, cultural integration, long-term capability	Improved performance on specific tasks, behavioral changes, skill acquisition
Examples in Engineering	Senior engineer guiding junior on career paths, navigating organizational politics, design philosophy, or industry trends	Helping an engineer improve code quality/review skills, prepare for a presentation, or adopt a new tool/methodology

What's In It For Me?

Now that we understand the differences between mentoring and coaching and why it matters in today's engineering world, let's look at the human side of it – WHY WOULD ANYONE WANT TO DO IT?

For that, I'll repeat what I said in the introduction.



The newer engineer benefits would include:

- **Faster Mastery of Real-World Skills:** Instead of learning through trial and error (often an expensive, frustrating, and possibly embarrassing process), new engineers would receive direct, practical guidance from seasoned professionals who have already solved complex problems in the field.

- **Preservation of Institutional Knowledge:** They would absorb the "unwritten rules," lessons from past failures, and nuanced judgment that textbooks and formal training rarely capture. (They make new mistakes – called “progress”- instead of repeating mistakes made by predecessors which doesn’t advance the culture.) This shortens the typical 5–10 year ramp-up period to full productivity.
- **Increased Confidence and Reduced Risk:** Having a trusted mentor provides a safety net for asking "dumb" questions (this is a critical factor), reviewing designs, and making decisions. This leads to fewer costly mistakes and faster professional confidence.
- **Career Development and Belonging:** A structured knowledge transfer program helps new engineers understand career paths, develop leadership skills, and feel genuinely valued. This dramatically improves engagement, job satisfaction, and long-term retention
- **Broader Perspective:** Exposure to the wisdom of senior engineers helps them develop better judgment, foresight, and engineering intuition.



Of course, the senior participants gain something, too:

- **Legacy and Sense of Purpose:** Many retiring engineers feel a quiet sadness watching their hard-earned knowledge walk out the door with them.

Mentoring through a structured knowledge transfer program allows them to leave a lasting legacy — knowing their experience will live on through the next generation. This often brings profound satisfaction and emotional closure.

- **Respect and Recognition:** Being formally asked to mentor signals that their career contributions are still highly valued. It honors their expertise instead of making them feel obsolete as retirement approaches.

"Anyone who stops learning is old, whether at twenty or eighty. Anyone who keeps learning stays young."
— Henry Ford. Great for stressing that knowledge transfer and personal growth keep teams (and individuals) innovative and vital.

- **Intellectual Stimulation:** Teaching forces seniors to organize and articulate their knowledge, often leading to new insights about their own work. Many discover they learn as much as they teach.

- **Reverse Mentoring Opportunities:** In a healthy program, older engineers are also exposed to new technologies, fresh approaches, and digital tools from their mentees. This keeps them engaged and relevant. (Also, learning a little humility can be a character enhancing experience.)
- **Smoother Life Transition:** Mentoring provides a meaningful way to gradually reduce workload while staying connected to the profession. It softens the identity loss many seniors experience when they fully retire.
- **Personal Fulfillment:** Numerous studies and personal accounts show that mentoring is one of the most rewarding activities for experienced professionals. It gives them a renewed sense of contribution and impact.”

Selecting Participants

These self-assessment tools help identify suitable participants, align expectations, expose knowledge gaps, and support better mentor-mentee (you may prefer “teacher-student” terminology because each will move into and out of the roles and they teach each other) matching. They are especially useful in organizations facing aging workforces, where the goal is effective tacit knowledge transfer.

Use these as a beginning and – remembering the three iterations I discovered in my own work – modify them as needed until their fit your culture comfortably. Also, you don’t always have to have a 1:1 matching – if a senior can work with two or more effectively, go for it! (Remember, this is a trial program and you’re working out the kinks as you go along.)

Example 1: Self-Assessment Questionnaire for Junior Engineers



Purpose: Help early- to mid-career engineers reflect on their current state, goals, and readiness for mentoring.

Instructions: Rate each item on a scale of 1–5 (1 = Strongly Disagree / Very Low, 5 = Strongly Agree / Very High). Add comments for key insights.

Section A: Technical & Professional Competencies

Statement	Rating	Comments
I have clear gaps in domain-specific technical knowledge (e.g., system architecture, troubleshooting legacy systems, regulatory nuances).		
I regularly apply engineering best practices but want deeper intuition on trade-off decisions.		
I am comfortable with modern tools/methods (e.g., simulation software, agile, AI-assisted design) but struggle integrating them with established practices.		
I can clearly articulate complex technical ideas to non-engineers.		

Section B: Career & Personal Development

Statement	Rating	Comments
I have well-defined short-term (1–2 years) and long-term (5+ years) career goals.		
I need guidance on navigating organizational politics, stakeholder management, or career progression.		
I am proactive in seeking feedback and learning from failures.		
I prefer hands-on learning (shadowing, projects) over purely theoretical discussions.		
I can receive objective criticism about my performance without taking it personally.		

Section C: Mentoring Expectations

Statement	Comments
What are your top 3 expectations from a mentor? (e.g., technical advice, career sponsorship, introductions)	
How much time per month are you willing to commit? (e.g., 4–8 hours)	
Describe a recent engineering challenge where you wish you had more experienced guidance.	
What is one area of tacit knowledge (undocumented “know-how”) you most want to gain?	
Potential concerns or barriers to participating (time, openness to feedback, etc.)	
Why would someone want to invest their time and knowledge mentoring you?	

Scoring Guidance: Total score isn’t definitive, but lower ratings in competencies + clear goals indicate strong mentoring candidates

Example 1: Self-Assessment Questionnaire for Senior Engineers



Purpose: Help experienced engineers evaluate their suitability, willingness, and capacity to serve as mentors, especially those nearing retirement.

Instructions: Rate 1–5 and reflect. ((1 = Strongly Disagree / Very Low, 5 = Strongly Agree / Very High). Add comments for key insights.)

Section A: Experience & Knowledge Depth

Statement	Rating (1-5)	Evidence / Stories (Add additional pages if needed)
I have deep expertise in areas critical to our organization (e.g., legacy systems, failure modes, design philosophy).		
I can articulate why certain decisions were made in past projects (trade-offs, context, risks).		
I have successfully solved complex, ambiguous engineering problems over my career.		
I stay reasonably current with emerging technologies and industry trends.		

Section B: Mentoring & Interpersonal Skills

Statement	Comment (Attach additional pages if needed)
I enjoy sharing stories and lessons from my experiences (successes and failures).	
I am comfortable using coaching techniques (asking questions) rather than always giving direct advice.	
I can provide constructive feedback in a supportive way.	
I am patient with different learning styles and generational perspectives.	

Section C: Commitment & Availability

Statement	Comment (Attach additional pages if needed)
How many years of experience do you have in relevant engineering domains?	
How much time can you realistically commit per month? (e.g., 4-8 hours)	
Are you open to phased retirement or part-time mentoring roles?	
What do you hope to gain from being a mentor? (e.g., legacy building, staying engaged, fresh perspectives)	
Any constraints or preferences (e.g., specific technical areas, in-person vs. virtual)?	

Example 2: Quick Matching & Compatibility Tool

Use this tool for both during the application phase. It is a self-reflection tool that will give you more information about matching participants.

Rate your preferences (1–5 scale 1 = Strongly Disagree / Very Low, 5 = Strongly Agree / Very High). Add comments for key insights.):

1. Using a structured vs. flexible meeting cadence
2. Stressing technical depth vs. leadership/career focus (which one?)
3. Frequency of direct advice vs. guided self-discovery (which preferred?)
4. Project-based mentoring vs. regular check-ins (which preferred?)
5. Interest in reverse mentoring (juniors teaching seniors on new tools) (1-5 scale)

Reflective Prompts (open-ended and attach additional sheets if needed.):

1. What is one piece of tacit knowledge (knowledge gained from experience, not documented in books or manuals) or lesson you would most like to share/receive?
2. Describe your ideal mentoring relationship in one paragraph.
3. What past experience (positive or negative) have you had with mentoring or knowledge sharing?

Implementation Tips for the Program Administrator

- **Format:** Digital (Google Forms, Microsoft Forms, or HR platform) for easy analysis and matching.
- **Frequency:** Initial self-assessment at signup; follow-up every 3 months initially - longer later.
- **Usage:** Combine with manager input and interviews for final selection. Aggregate anonymous data to identify common gaps for group training.
- **Engineering-Specific Customization:** Tailor competency sections to your domains (e.g., mechanical, electrical, software, systems, safety).
- **Benefits:** Increases program success rates, sets realistic expectations, and surfaces high-potential pairs for tacit knowledge transfer.

Example 3: An In-depth Informational Gathering Tool

These questions designed for both partners – the junior and senior participants – help to promote mutual understanding, identify expectations, reveal teaching/learning styles, and help evaluate their program fit.

1. What are your main goals or what do you hope to gain from participating in this knowledge transfer program?

(Reveals motivation, alignment with program objectives, and whether they're looking for growth, legacy-building, skill development, or networking.)

2. Describe a time when you successfully learned something complex from someone more experienced (for juniors) or helped someone learn a difficult concept/skill (for seniors). What made that experience effective?

(Highlights past successes, preferred learning/teaching methods, and gives insight into what works for them personally.)

3. What is your preferred way to share or receive knowledge? (e.g., hands-on projects, one-on-one mentoring, documentation, workshops, shadowing, regular check-ins, etc.)

(Uncovers style compatibility—some seniors love storytelling while juniors may prefer practical demos.)

4. What do you believe are the biggest challenges or gaps in how knowledge is currently transferred in our company (or in your experience)? How could this program help address them?

(Shows awareness of real issues, critical thinking, and commitment to improving the process.)

5. How do you handle situations where there's a difference in opinion or approach on technical topics? Can you share an example?

(Assesses openness, humility, respect, and conflict resolution—critical for productive mentor-mentee relationships.)

6. What excites you most about working with someone more/less experienced than you, and what concerns you (if anything)?

(Directly exposes potential stereotypes, fears, or enthusiasm. Seniors often worry about time commitment; juniors about being micromanaged.)

7. On a scale of 1 (low)–10 (high), how important is it for you to have structured milestones/check-ins versus a more flexible/organic approach? Why?

(Helps match pairs with compatible expectations around program structure and accountability.)

8. What would make this program a great success for you six months from now?

(Encourages visualization of outcomes and reveals their definition of value—technical skills, confidence, relationships, documentation, etc.)

Tips for Implementation:



- Ask these previous questions during application/interview or in a matching survey.
- Have participants review anonymized or summarized responses from potential matches (with permission) to improve pairing.
- Use the answers to identify red flags (e.g., low motivation, rigid styles) or strong fits (complementary goals and methods).
- You can add a short self-assessment on communication style, availability (hours per month), or preferred technical domains.

Example 4: Matching Methods with Knowledge



A mentor that I had early in life, my Dad, often told me that *“if your only tool is a hammer, then every problem looks like a nail.”*

It wasn't until I got into teaching that I really understood what that meant as I realized that different students or different kinds of knowledge require different instructional approaches. I couldn't teach the same way every time – my teacher's toolbox needed more than a hammer.

So here are a few different approaches to try with different situations. I want you to have as many tools available as possible.

1. Direct Interaction and Observation Methods

"Tell me and I forget, teach me and I may remember, involve me and I learn." — Benjamin Franklin (or variations attributed to Confucius/Xunzi). Stresses hands-on, experiential knowledge transfer over passive methods—ideal for engineering.

Most Effective for Tacit Elements resulting from hands-on experience in mentoring, apprenticeships, and job shadowing ("follow me and see what I do")

Pair experts with juniors for extended periods. The junior observes workflows, decision-making, problem-solving in real time, and asks questions (or the teacher asks, "why do you think I'm doing it this way?"). This preserves nuances *like "why I'm doing this instead of that."*

Strength: High-fidelity transfer of intuition and context.

Limitation: Time-intensive; scales poorly – only a few learners at a time.

Tip: Use structured programs with clear goals, rotation, and documentation of key insights.

Observation and "Show Your Work": Experts demonstrate tasks while narrating thought processes. Video recordings help.

The best teaching method for this is "TriPLICATION" – doing something in three distinct steps:

- (The teacher) "I do the skill so you (student) can watch and **I explain** ("I say") what I'm doing as I do it."
- "I do the skill again but you say (explain) what (and why) I'm doing it."
- "You do the skill and you say what (why) you're doing."

"I do, I say; I do, you say; you do, you say" TriPLICATION is effective for physical or procedural skills.

Communities of Practice (CoPs): Groups of people with shared expertise meet regularly (in-person or virtual) to discuss challenges, share stories, and solve problems collaboratively.

2. Elicitation and Storytelling Techniques, Structured Interviews, and Cognitive Interviewing:

Use open-ended questions, scenario-based prompts ("*Walk me through how you handled that crisis*"), and follow-ups to probe intuitions. Record these sessions and build a library.

NOTE: Be sure to ask for a specific outcome to keep them on track and prevent rambling in their stories. "*Tell me just what you did to shave those 2 months off the project.*"



Best for: Retiring experts or knowledge holders.

Storytelling Sessions and After-Action Reviews (AARs): Experts recount real experiences, lessons learned, and “war stories.” AARs review projects post-completion: what happened, why, and what to do differently.

Exit Interviews and Knowledge Handover: Systematic sessions before departure, focusing on critical decisions, undocumented tips, and networks.

3. Collaborative and Social Methods Workshops, Training, and Cross-Functional Teams:

Facilitated sessions where experts share via role-playing, simulations, or group problem-solving.

Knowledge Mapping: Visualize expertise networks, key processes, and dependencies to identify gaps and holders.



4. Digital and Technology-Enabled Capture

Video/Audio Recording and Transcription: Capture demonstrations, interviews, meetings, or casual conversations. AI tools transcribe, summarize, and tag content for searchability.

Collaboration Platforms: Microsoft Teams, Zoom, etc., with recordings, chats, and integrated knowledge bases.

Knowledge Management Systems (e.g., Bloomfire, wikis, intranets): Store videos, annotated guides, forums, and Q&A. AI enhances search and recommendations.

Emerging AI Approaches (as of 2025–2026): AI avatars/digital humans that interact conversationally, drawing on captured expertise.

Tools that analyze workflows, prompt for rationales, or use Large Language Models (LLMs) to externalize ideation. Voice capture + AI summarization for turning discussions into assets.

5. **Process and Cultural Methods & Lessons Learned Databases: Standardized templates for projects. Phased Retirements:** Keep experts part-time as mentors or consultants.

Knowledge-Sharing Culture: Incentives, recognition, leadership modeling, and integration into performance reviews.

Challenges and Best Practices Tacit knowledge is sticky and context-dependent — Over-codification can lose nuance. Combine methods (e.g., video + mentoring + searchable repository).

Barriers: Time, reluctance to share (fear of replacement), lack of trust, or poor facilitation.

Success Factors: Start with high-risk areas (retirement cliffs in critical roles).

Prioritize: Identify key knowledge holders and critical items first.

Measure: Track reuse of captured knowledge, reduced errors, or faster onboarding.

Blend human + tech: Technology amplifies but doesn't replace relationships.

For aging workforces: Begin transfers 2–5+ years in advance via intergenerational pairing.

Overall Effectiveness: Human-centric methods (mentoring, storytelling) excel at deep transfer, while digital tools provide scale and accessibility.

Organizations that treat this as an ongoing cultural practice—rather than a one-off project—retain the most value.

In high-stakes fields (aerospace, healthcare, manufacturing, engineering), combining these has proven to mitigate retirement risks significantly.

Reverse KT Strategies



Transferring knowledge from **juniors to seniors** (also called “Reverse Mentoring”) flips the traditional model: early-career or younger engineers teach more experienced/senior engineers on specific topics where they hold an advantage.

This is particularly powerful in engineering, where rapid technological change, new tools, sustainability practices, and fresh perspectives on problem-solving are critical.

Why Reverse KT Matters in Engineering

- It bridges the generational knowledge gap and accelerates adoption of new technologies (AI tools, BIM software, simulation platforms, drones/LiDAR, sustainable design methods).
- It improves retention of younger talent by making them feel valued and heard.
- It can enhance innovation, diversity of thought, and organizational agility.
- It helps senior engineers stay current without feeling "behind," while building humility, adaptability, and stronger cross-generational relationships.
- It can support broader goals like licensure preparation, ethical decision-making in emerging areas (e.g., climate resilience), and reducing resistance to change.

Real-world examples include GE (pioneered by Jack Welch for internet/tech skills), P&G (digital transformation and AI applications), and various engineering/tech firms using it for software proficiency and sustainability.

Key Benefits for Engineering Organizations

- **Knowledge Transfer:** Juniors teach seniors about emerging tools; seniors share institutional knowledge, project judgment, and client management.
- **Cultural Impact:** Builds psychological safety, reduces hierarchy barriers, and improves collaboration on multidisciplinary projects.
- **Business Outcomes:** Better retention, faster innovation, and leaders who make more informed technical decisions.

Practical Reverse KT Strategies

Here are actionable strategies tailored for engineering environments:

1. **Define Clear, Narrow Objectives** Focus on specific skills rather than vague "learn from each other" such as:
 - Mastering new design software or AI-assisted analysis.
 - Applying sustainability metrics or ESG principles in projects.
 - Understanding Gen Z communication preferences or hybrid team dynamics.
 - Fresh perspectives on risk assessment or failure modes.
 2. **Thoughtful Matching** Pair based on complementary strengths and compatibility (not just hierarchy).
 - Use skills inventories or short surveys like we have in the "Selecting Participants" section starting on page 22.
 - Avoid direct supervisor-subordinate pairs to reduce power dynamics.
 - Consider diversity (background, gender, ethnicity) for broader perspectives.
 3. **Structure the Relationship**
 - Create a simple Reverse KT Agreement (goals, meeting cadence, confidentiality, duration — typically 6-12 months).
 - Schedule regular meetings (e.g., 45-60 minutes every 2-4 weeks).
 - Use a mix of formal sessions and informal "shadowing" or joint problem-solving.
 - Encourage project-based learning: Have the junior lead a tool demo on a real (low-risk) project component.
- "It's what you learn after you know it all that counts."** — John Wooden (also associated with Harry S. Tuman and Earl Weaver). This is the perfect anchor quote for your program—it underscores intellectual humility and the need for ongoing growth even (especially) among experts.
4. **Train Both Parties**
 - Seniors (as students): Humility training, how to receive feedback, setting ego aside.
 - Juniors (as teachers): Communication skills, structuring sessions, giving feedback upward without intimidation.
 - Provide templates for agendas and progress tracking.

5. Incorporate Engineering-Specific Activities

- Tool walkthroughs (e.g., new finite element analysis features).
- Joint design reviews where the junior challenges assumptions with fresh data.
- Case studies on recent project failures/successes viewed through a modern lens.
- Review engineering-related ethics case studies from www.nspe.org
- Reverse shadowing: Junior leads a client meeting or site visit discussion.

6. Measure and Sustain Success

- Track metrics: Adoption rate of new tools, project improvements, participant feedback, retention rates.
- End with a wrap-up session and option to continue informally.
- Share success stories internally to build momentum.

Generational Challenges and How to Overcome Them

Trying to get two former strangers to work together in a stressful and structured engineering environment will not be the easiest thing you have ever tried to do but it is possible.

Here are some challenges you can expect and suggestions on dealing with them. (Don't worry - you will learn more about dealing with participants in your fourth or fifth iteration of your program!)

Trust is Historic – *people must learn through experience they can trust you.*

No amount of just saying, "trust me" is going to change that.

- **Resistance/Ego:** Seniors may feel threatened. → Frame it as mutual learning and leadership development. This is another reason to be careful in the selection of your first two participants.
- **Power Dynamics:** Juniors hesitant to "teach the boss." → Emphasize reciprocity and leadership support.
- **Time Constraints:** Busy project schedules. → Keep sessions short and focused; integrate into existing workflows.
- **Mismatch:** Poor pairing. → Allow opt-outs or re-matching after initial sessions.

Implementation Tips for Engineering Firms

- Start small (one department or pilot group of 6-10 pairs).
- Get visible leadership buy-in (e.g., directors publicly participating).
- Integrate with existing KT programs rather than replacing them.
- Leverage technology: Shared digital notebooks, recorded demos, collaboration platforms.

Potential Engineering KT Scenarios

So far, we've been down in the details looking at the nuts and bolts of transferring knowledge and it's easy to lose sight of the big picture – just what could this look like in a real engineering setting?

Here are three situations that will help you understand the general outcome with a few specific deliverables.

AI Tool Adoption



A 28-year-old engineer teaches a 55-year-old Project Manager how to use and apply an AI-powered structural analysis plugin.

Outcome of Good Approach: By showing some humility and a willingness to listen and learn for three sessions, the senior PM incorporates the tool on a bridge rehabilitation project, reducing analysis time by 40% while maintaining full manual verification for critical elements.

Poor Approach: Or, the senior dismisses the tool as “some newbie with her black box magic” and refuses to learn, causing the team to fall behind competitors.

Sustainability Perspectives

A Gen Z engineer teaches a senior civil engineer how younger clients evaluate projects using carbon footprint metrics and circular economy principles.

Result: The senior engineer updates proposal templates and wins a major municipal project by addressing these concerns proactively.

Communication Style



Junior shows senior how to use short-form video updates instead of long emails for field team coordination *resulting in improved response time and fewer misunderstandings on construction sites.*

Measuring Success and ROI

Here are examples of ways to measure and report your ROI. When executives want to see some outcomes from your trial period, these are types of measurements you can offer.

Naturally the nature of your project, time available, and cooperation of those involved will determine the best metrics for you to use.

Category	Metric	How to Measure	Target / Benchmark	Data Source
Productivity	Task completion rate	Tickets/points completed per week	+15-30% for mentees	Jira, Azure DevOps
	Ramp-up time to full productivity	Weeks to reach 80% of senior output	Reduce by 4-8 weeks	Self-report + manager review
Quality	Defect density / escape rate	Bugs per 1k lines or per feature	-20-40%	Bug tracking system
Knowledge Transfer	Knowledge reuse rate	% of solutions reused from mentor sessions	>60%	Wiki edits, PR references
Innovation	Ideas implemented	Number of process/tools improvements adopted	+2-5 per pair/year	Idea tracker / retros
Business Impact	Project delivery acceleration	% reduction in cycle time for key deliverables	-10-25%	Project management tools
Retention	Mentor/Mentee retention rate	% staying 12+ months post-program	>85%	HR system
Cost Savings	Reduced external training/support	\$ saved on consultants, courses, rework	Track \$ value	Finance + training logs



And now, it's time for a small confession. Since I've been retired from actively teaching and away from the engineering world for some time, I'm not as current with terms, products, and concepts as I once was. However, I have been smart enough to realize what AI can do if you ask the right questions!

So I asked Grok (Elon Musk's AI creation) to explain parts of this as if I were 10 years old and I will pass that on in this section. Not that I think my readers (engineers) couldn't understand things more complex but keeping things as simple as possible makes it easier to adapt to your situation.

Productivity Gains

How we measure it: (Be sure to use the same measurement instrument for the before and after measurements.)

- Count how many "jobs" (tasks or tickets) the person finishes in a week.
- See if they finish work faster than before.
- Example: "Sarah used to finish 8 tasks a week. After mentoring, she finishes 12. That's a big productivity gain!"

We just compare the numbers before and after the mentoring.

Quality Improvements / Defect Reduction

How we measure it:

- Count how many mistakes (bugs or problems) happen.
- Example: "Before, the team had 20 mistakes every month. After mentoring, only 10 mistakes. That's quality improvement!"

Fewer mistakes = better quality. We track the number of bugs and how bad they are

Knowledge Retention & Reduced Ramp-up Time

How we measure it:

How many weeks it takes the new person to work at full speed.

- **Test them later:** "Can you still explain and use what you learned?" (A verbal explanation or a physical demonstration is proof of understanding.)
- Example: "New engineers usually take 10 weeks to be good. With mentoring, they only need 5 weeks — and they still know it 6 months later."

We measure time saved + how well they remember.

Innovation / Idea Generation

How we measure it:

- Count how many new ideas the person suggests.
- Count how many of those ideas actually get used.
- Example: "Before, the team got 2 new ideas a month. After mentoring, they get 6 new ideas and use 4 of them."

It's like counting new inventions or smart shortcuts.

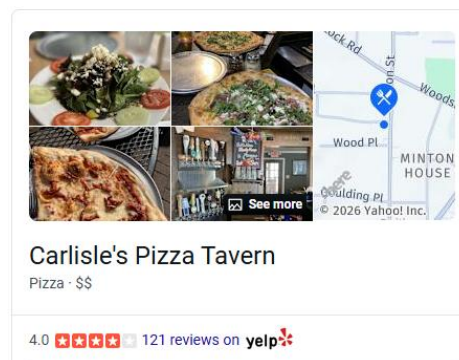
Retention & Engagement

How we measure it: (Human Resources may already be measuring something close to this. Be sure to check with them first so you don't "reinvent the wheel" unnecessarily.

- Do people stay in the company or team longer? (How much longer – can you tie it to this program?)
- Do they say they're happy and excited about their work? (Simple surveys like "On a scale of 1-10, how much do you like your job now?")
- Example: "Before the program, 3 people left the team. After, almost nobody left and everyone feels more excited."

We look at who stays and how happy they feel.

Measuring Perceptions



We have become used to and have a high-reliance on the 5-star scoring system which has become a universal way to rate something.

Whether we're looking for great pizza, a dentist, or a plumber, our first question is "how are they rated?"

We trust it in our private lives but how much will we trust it in a business? More specifically in a highly technical, data-driven enterprise like an engineering company?

The science behind the 5-star system is a trusted tool from the worlds of Marketing and Psychology called a Likert Scale (from the developer, Rensis Likert, in 1932). It presents a positively worded statement about a topic of interest and asks for the subject's agreement along a 5-7 scale ranging (usually) from 1= Disagree strongly 2= Disagree 3 = Neutral/No opinion either way 4= Agree 5= Agree Strongly.

You get a vertical score for the subject by getting a simple average of all of their responses on one survey (like the 4.0 on the pizza restaurant in the picture). But you can also parse details scoring horizontally across many surveys to discover trends about the topic. (You should use this strategy in designing the surveys for your program. There are examples below.)

For example, let's say the 4.0 overall score represents the "dining experience" at the restaurant but the survey breaks the overall vague "dining experience" concept into very specific components such as first impression of the outside, parking, service quality, and food.

Collectively, those components made up our dining experience. (*We have more statements about the food than the parking because it is so much more important.*)

Now your survey looks like this – and you just average the scores:

First impression of outside

- *"The restaurant appeared inviting with well-maintained landscape and exterior" 5*
- *"There was customer activity that made it look inviting." 5*

Parking

- *"The parking is easy to access and safe to use." 5*
- *"The parking area is well-lit and clean." 2*

Service Quality

- *"I (we) were promptly and cheerfully greeted upon arrival." 5*
- *"Seating was comfortable and uncrowded." 5*
- *"The server was appropriately attentive – neither too much nor too little. (Please underline 'too much' or 'too little' as reflected in your score)" 4*

Food

- *“The taste and flavor of the food were exceptional.” 4*
- *“The components of the meal tasted fresh.” 5*
- *“The hot and cold items were served at the appropriate temperatures.” 5*
- *“The meal presentation was very appealing.” 5*
- *“The portion sizes were generous.” 5*
- *The price (without gratuity) accurately reflected the value of the experience.” 4*

The score of this survey is 59/13 (questions) = **4.5** (the **VERTICAL SCORE** from 1 customer). Looks like they had a very good dining experience at the restaurant. (But they need to work on the parking area -#2- which is a **HORIZONTAL SCORE** to be collected across many surveys for an aggregate score.)

A project manager can track the horizontal scores in a Likert survey to fine-tune issues that may pop up without bothering the entire organization’s momentum.

Likert Survey Design Tips

Core Design Tips for Higher Reliability

- Use 5-point or 7-point scales — These are the sweet spot for reliability. 5-point is simpler for busy engineers; 7-point gives more nuance.
- Always use multiple questions per topic — Never rely on a single Likert item. Combine 3–5 related questions and average them. This dramatically improves reliability.
- Clear, simple language — Use short sentences (reading level ~8th grade or lower). Avoid jargon unless everyone understands it.
- Balanced scales — Equal number of positive and negative options. Include a neutral midpoint (e.g., Neither Agree nor Disagree).
- Consistent wording & anchors — Keep the scale labels the same throughout the survey when possible.
- Avoid double-barreled questions — Don’t ask two things in one question.
- Avoid leading or biased wording — Don’t push the respondent toward an answer.
- Pilot test — Run the survey with 5–10 people first and ask if anything was confusing.
- Add an open comment box after each section — This helps explain extreme scores and improves overall data quality.

Recommended Likert Scale (for Mentoring Surveys) 5-point (easiest):

1 = Strongly Disagree

2 = Disagree

3 = Neither Agree nor Disagree

4 = Agree

5 = Strongly Agree

Topic	Bad Question (Unreliable)	Good Question (More Reliable)	Why the Good One is Better
Overall Satisfaction	"The mentoring program is great, right?"	"I am satisfied with the mentoring program overall."	Neutral, not leading
Productivity	"Mentoring helped me get way more work done"	"Since starting mentoring, I complete tasks faster than before."	Clear timeframe & single idea
Knowledge Transfer	"My mentor taught me a lot of useful stuff"	"My mentor helped me understand technical concepts I struggled with."	Specific & measurable
Confidence	"Do you feel better now?"	"I feel more confident debugging complex issues."	Specific skill area
Double-barreled (Bad)	"My mentor is helpful and available when I need him."	Split into two: 1. "My mentor gives helpful advice." 2. "My mentor is available when I need support."	One idea per question
Ramp-up / Learning	"Mentoring made onboarding fast"	"The mentoring program shortened the time it took me to become productive."	Clear and specific
Recommendation	"Everyone should do this program"	"I would recommend this mentoring program to other engineers."	Personal & clear

Here are a few Likert-style questions that may be useful in your KT program about its perceived Impact on skills (Rate each statement)

1. *I can solve technical problems faster because of the mentoring sessions.*
2. *My code quality has improved since working with my mentor.*
3. *I feel more confident when explaining my technical decisions.*
4. *The mentoring has helped me learn new tools or techniques.*
5. *I now handle complex tasks with less help than before.*

Maybe follow-up the survey with an open-ended question: “*What is one specific skill or knowledge area where you noticed the biggest improvement?*” This may uncover nuggets of success that you never anticipated.

A sample survey focusing on technical and professional growth (Juniors)

1. *I can solve complex engineering problems faster than before.*
2. *My confidence in making design/trade-off decisions has increased.*
3. *I have absorbed important tacit/unwritten knowledge.*
4. *I make fewer technical mistakes than I did 6 months ago.*
5. *I feel more prepared for my next career milestone (PE, lead role, etc.).*

A survey for seniors about their legacy & engagement

1. *I feel my knowledge and experience are being preserved.*
2. *Mentoring has given me renewed purpose and satisfaction.*
3. *I have learned valuable new skills/tools from my junior partner (reverse KT).(Consider having a space here where they specify what they were.)*
4. *I feel more respected and valued by the organization.*

A survey about program effectiveness for both partners (Both)

1. *Overall satisfaction with the KT program (1–10)*
2. *Likelihood to recommend the program (1-10 NPS)**
3. *Estimated monthly time invested*
4. *Perceived value of time invested (1–5)*

*Since engineers love numbers and calculations, here’s a frequently-used one from marketing that you may not know about called the **Net Promoter Score (NPS)**

How NPS is Calculated:

- **Promoters** (score 9–10): People who love the program and will actively recommend it.
- **Passives** (score 7–8): People who are satisfied but not enthusiastic.
- **Detractors** (score 0–6): People who are unhappy or neutral-to-negative.

NPS Formula: $NPS = \% \text{ of Promoters} - \% \text{ of Detractors}$

Example: 50 people answer the survey. 30 give 9–10 → 60% Promoters; 10 give 0–6 → 20% Detractors; $NPS = 60\% - 20\% = +40$ (You’ll have to decide your threshold for an acceptable NPS)

Overcoming Resistance



Not everyone is going to share your enthusiasm about your KT program. In fact, there may be some resistance from people who feel it will “rock the boat” or threaten their jobs (WHAT? You expect me to train my replacement??)

If it comes, don’t be discouraged. This is why we insisted on making your first choice of pairs (assuming a 1:1 matching in your organization) ones that will make the program (and you) look good.

Here are some of the most likely comments and/or resistance you will have with some response strategies.

- **“I don’t have time”** → Make sessions short and high-value (25–45 min max, bi-weekly). Tie mentoring directly to current work (e.g., “Let’s review this ticket together” instead of generic meetings).
- **“I don’t need help”** (Ego/Independence) → Frame it as peer knowledge exchange, not “teaching.” Use reverse mentoring (juniors teach seniors new tools) to reduce status gap.
- **“This is extra admin work”** → Minimize paperwork. Use existing tools (Jira comments, quick Notion templates, or 2-minute session logs).
- **Manager pushback** → Show early wins with metrics (productivity or defect reduction) and get leadership to visibly support it (e.g., “Mentoring time counts toward goals”).

Run a quick anonymous survey early: *“What would make this program easier or more valuable for you?”*
Act on the top 2–3 responses visibly.

Maintaining Engagement Over Time Engagement often starts high and drops after 6–8 weeks.

Proven tactics:

- **Clear wins every session** — End every mentoring session with a concrete action item and quick *“What’s one thing you’ll try this week?”*
- **Gamification lite** — Track “knowledge points,” badges for completed challenges, or public (opt-in) shout-outs in team channels.
- **Varied formats** — Mix 1:1s with group sessions, code walkthroughs, “lunch & learn,” or shadowing.
- **Accountability partners** — Pair people and have them share one win per month in a lightweight team sync.
- **Personalize goals** — Link mentoring to individual career goals (promotion, new skills, side projects).
- **Regular pulse checks** — Use the short Likert surveys we discussed earlier every 4–6 weeks. Ask: *“On a scale of 1–5, how valuable were recent sessions?”*

Rule of thumb: If engagement drops, reduce frequency before adding more — quality beats quantity.

Handling Program Drift

Program drift happens when activities become unstructured or drift away from original goals.

Prevention & correction:

- **Define success clearly upfront** — Document 3–5 key outcomes (e.g., reduce ramp-up time by 30%, improve code quality metrics).
- **Use light structure** — Mentoring charter or simple playbook: goals, meeting cadence, expected outputs.
- **Quarterly alignment reviews** — 30-minute check with mentors, mentees, and managers: *“Are we still on track? What needs adjustment?”*
- **Dashboard monitoring** — Track the quantitative metrics (productivity, defects, knowledge artifacts). If numbers flatline, investigate quickly.
- **Refresh pairs** — After 6–9 months, consider rotating or graduating pairs while keeping some continuity.

Keeping It Alive Beyond the Pilot Phase The biggest drop-off usually happens when moving from pilot to ongoing program.

Sustainability playbook:

- **Embed into existing processes** — Make mentoring part of onboarding, performance goals, or promotion criteria (e.g., “*Senior engineers are expected to mentor*”).
- **Build internal champions** — Identify 2–3 enthusiastic mentors and give them light leadership roles or recognition.
- **Show ROI visibly** — Share success stories + numbers (e.g., “*This pair reduced defects by 35% and delivered features 3 weeks early*”) in leadership updates.
- **Secure budget & resources** — Even small ongoing budget for tools, occasional lunches, or swag helps signal importance.
- **Create a self-sustaining loop** — Turn successful mentees into future mentors. This creates a knowledge flywheel.
- **Annual refresh** — Treat it like any product: run a retrospective, update the program based on feedback, and re-launch with improvements.

Long-term health check questions (ask every 6 months):

- *Are we still seeing measurable gains in productivity/quality?*
- *Do participants still rate it 4+ out of 5?*
- *Are new people joining voluntarily?*
- *Is it running with minimal central coordination?*

Common Pitfalls to Avoid

- *Over-administering (too many forms)*
- *Forcing participation instead of attracting it*
- *Letting sessions become unstructured venting*
- *Forgetting to celebrate wins*

Final advice: Start small, prove value fast, then institutionalize. Treat the mentoring program itself like an engineering project — iterate based on data and feedback