Leadership Guidelines for an Agile Team

Part 1: Preparation and Development

Paul Alfred Elling

Contents

This book is dedicated to my family: my wife, Janelle and my son, Isaac.

Paul Alfred Elling

Introduction

Not enough education is provided to people, including students, about the subject of leadership. It can be considered art and science and has been practiced by people in various situations and organizations throughout the world and through the course of history. Leadership is available to anyone from any walk of life. It is not limited to gender, race, religion, physical attributes, or economic means. Rather, leadership is about galvanizing oneself and others to accomplish something greater than can be achieved by individual efforts. It is the goal of this book to provide guidelines of leadership that can be learned and practiced accomplishing the goals of an organization.

“Leadership can be defined as the art, science, or gift by which a person is enabled and privileged to direct the thoughts, plans, and actions of others in such a manner as to obtain and command their obedience, their confidence, their respect, and their loyal cooperation. Leadership is the profession of the officer in which proficiency can only be obtained through a constant study of leadership principles and practice in applying them in day-to-day relationships with juniors, seniors, and peers. The most successful leaders have a keen understanding of human behavior and know how to get the most from their followers.” (p. 1 – 3, Navy)

“Leadership is not about a position or title, as many young people think. It is about choices you make throughout your life—with the goal of making the situations and places you find better because you were there. Great leadership is not about making the leader look good but about how individuals use leadership in service to others to make the people and groups around them better. Leadership is the art of mobilizing others to want to struggle for shared aspirations.” (p. XV – 1, Kouzes and Posner)

I obtained an MBA in Strategic Leadership from Amberton University because I wanted to learn what it takes to lead. I have learned about leadership from my career experiences as well. What I have realized over time is that learning about leadership happens throughout life as a never-ending process.

Members at all levels of an organization can lead. The role of the leader is to define reality and give hope. Followers can be leaders when they embrace their leaders' goals and when they lead by example. Leadership begins when leaders seize opportunities, and it is all about behavior. "A leader is someone who takes responsibility. Leadership is born when we become active rather than passive." (p. 5, Sacks) Leaders must have the courage to be themselves and not someone else. "The core task of leadership is to create the conditions for other people to thrive." (p. 9, Frei and Morriss) "Leadership is the influencing process of leaders and followers to achieve organizational objectives through change." (p. 6, Lussier and Achua)

These leadership books will focus on the intangibles, the fundamentals, and the principles necessary for an Agile team to succeed. The first book will focus on the process, or the building blocks, that are needed. The second book will focus on development, or the fundamental skills, that are needed. The third and fourth books will focus on the principles, or the bedrock, that are needed.

Part 1: Process

Chapter 1: Preparation

Preparing to be part of an Agile team can begin as early as college, when the opportunities to be part of different groups are more readily available to individuals. College campuses offer a variety of groups which students can join. These groups include student government, fraternities related to college majors, cultural interests, community organizations, and many others. They are networking opportunities that allow students to learn what they care about as well as meet other students and faculty members who share their interests.

Student Leadership

Kouzes and Posner offer The Five Practices of Exemplary Leadership to help anyone, including students, learn to lead. The Five Practices of Exemplary Leadership are:

- Model the Way (p. 10, Kouzes and Posner)
    - “Understand who you are and what your values are”
    - “Find your voice”
    - “Help identify and affirm the shared values of the group”
    - “Words and actions must be consistent”
    - “Set the example by aligning actions with shared values”
- Inspire a Shared Vision (p. 10, Kouzes and Posner)
    - “Leaders envision the future by imagining exciting and ennobling possibilities”
    - “Enlist others in a common vision by appealing to shared aspirations”
- Challenge the Process (p. 11, Kouzes and Posner)

    - “Search for opportunities by seizing the initiative and by looking outward for innovative ways to improve”
    - “Consistently generate small wins and learn from experience”
- Enable Others to Act (p. 12, Kouzes and Posner)
    - “Foster collaboration by building trust and facilitating relationships”
    - “Strengthen others by increasing self-determination and developing competence”
- Encourage the Heart (p. 13, Kouzes and Posner)
    - “Recognize contributions by showing appreciation for individual excellence”
    - “Celebrate the values and victories by creating a spirit of community”

First, leaders must clarify values by asking themselves who they are when they want people to follow them. “Leaders are expected to be able to speak out on matters of values and conscience, and to be clear about what matters to them.” (p. 21, Kouzes and Posner) Know what to speak about and the beliefs you stand for. “To earn and sustain personal credibility, you must first be able to clearly articulate deeply held beliefs. You must freely and honestly choose the principles you will use to guide your decisions and actions.” (p. 21 – 22, Kouzes and Posner) Second, leaders need to find their voice by asking themselves what their leadership philosophy is. “You need to reaffirm your leadership philosophy on a daily basis. Find your authentic voice—the most genuine expression of who you are.” (p. 22 – 23, Kouzes and Posner) Express yourself in your own words. To find your voice (p. 23):

- “Discover what you care about, what defines you, and what makes you who you are.”
- “Explore your inner self.”
- “Lead according to the principles that matter most to you.”

Next, leaders must affirm shared values and set the example. "Leadership is also about the values of those you lead. Shared values are the foundation for building productive and genuine working relationships." (p. 28 – 30, Kouzes and Posner) Leaders must give people reasons to care. As a leader, what you do is more important than what you say. Live the shared values by spending your time and attention wisely.

> "Exemplary student leaders understand and are attentive to language because they appreciate the power of words. Asking relevant questions not only to challenge others to connect their actions with the team's values but also to ask their colleagues and team members about how their own actions as a leader impact both the feelings of others and their performance. The best leaders are self-aware and socially aware." (p. 47 – 51, Kouzes and Posner)

Leaders teach others to model the values. "Look for opportunities to teach not just by your example, but also by taking on the role of teacher and coach." (p. 54, Kouzes and Posner) Rather than refrain from incidents, leaders confront them. "Critical incidents present opportunities for leaders to teach valuable lessons about appropriate norms of behavior and what really matters." (p. 55, Kouzes and Posner) Leaders take the opportunity to tell stories. "Stories are powerful tools for teaching people about what's important and what's not, what works and what doesn't, what is and what could be. Through stories, leaders define culture, pass on lessons about shared values, and get others to work together." (p. 57, Kouzes and Posner) Leaders reinforce through systems and processes. "All exemplary student leaders understand that you have to reinforce the fundamental values that are essential to building and sustaining the kind of culture you want." (p. 59, Kouzes and Posner)

Leaders envision a shared vision by envisioning the future. Exemplary leaders "envision the future and gaze across the horizon, seeing greater opportunities to come." (p. 70, Kouzes and Posner) Being forward-looking is an attribute that followers look for in leaders. A vision is the main message a leader wants to convey. Leaders need to reflect on their past, attend to their present, prospect the future, and express their passion. (p. 72, Kouzes and Posner) "A significant part of being a leader is developing a deep understanding of where things are going." (p. 77, Kouzes and Posner)

Leaders find a common purpose with their followers. People want to unite with their leaders on common goals. Listening to other people's aspirations helps the leader determine what the common goals are. "The best student leaders listen carefully to what other people say and how they feel. People commit to causes, not plans. People commit to something much bigger, something much more compelling than goals and milestones on a piece of paper." (p. 84 – 87, Kouzes and Posner) People will follow when there is a clear vision.

Leaders make the effort to enlist others. "People want their leaders to be inspiring." (p. 96)

> "When you communicate your vision of the future to your group, you need to talk about how they're going to make a difference, how they're going to have a positive impact on people and events. You need to show them how their long-term desires can be realized by enlisting a common vision. You need to speak to a higher meaning and purpose of your group's work. You need to describe a compelling image of what the future could be like when people join a common cause." (p. 98, Kouzes and Posner)

Leaders appeal to common ideals by doing the following (p. 98 – 104, Kouzes and Posner):

- "Connect to what's meaningful to others."
- "Take pride in being unique."
- "Align your dream with the people's dream."

“To enlist others, you have to help them see and feel how they are aligned with the vision.” (p. 104, Kouzes and Posner) Leaders animate the vision by doing the following (p. 106 – 116, Kouzes and Posner):

- “Use symbolic language.”
- “Create images of the future.”
- “Practice positive communication.”
- “Express your emotions.”
- “Speak genuinely.”

Leaders challenge the process by searching for opportunities and seizing the initiative. “Leadership demands altering the business-as-usual environment. The study of leadership is the study of how men and women guide others through adversity, uncertainty, and other significant challenges.” (p. 126, Kouzes and Posner) Leaders make things happen by doing the following:

- “To do your best as a leader, you must seize the initiative to change the way things are.” (p. 127, Kouzes and Posner)
- “New jobs and assignments are ideal opportunities for asking questions and challenging the way things are done.” (p. 128, Kouzes and Posner)
- “Leaders earn the respect of the people around them when they question the status quo, come up with innovative ideas, follow through with the changes they suggest, get feedback, understand their mistakes, and learn from failures. Leaders don’t wait for permission or specific instructions before jumping in. They notice what isn’t working, create a solution for the problem, gain buy-in from constituents, and implement the desired outcome.” (p. 129, Kouzes and Posner)

Leaders encourage initiative in others by doing the following:

- “Change requires leadership, and every person in the group, not just the leader, can come up with creative ideas and suggest improvements.” (p. 130, Kouzes and Posner)
- “Promote a can-do attitude.” (p. 132, Kouzes and Posner)
- “Leaders encourage initiative by providing visibility and access to role models.” (p. 132, Kouzes and Posner)

Leaders challenge with purpose by doing the following:

- “Leaders raise challenges to resolve and improve the situation.” (p. 133, Kouzes and Posner)
- “Leaders get people to see beyond the particular task or project and to focus on the meaning that it fulfills.” (p. 134, Kouzes and Posner)

Leaders exercise outsight, which is “the awareness and understanding of outside forces.” (p. 135, Kouzes and Posner) Leaders are open to innovation coming from anywhere. Look outside your own experience by moving outside your usual thinking patterns. Listen to and promote diverse perspectives. “Leaders must be receptive to new ideas if they are to challenge the process effectively.” (p. 138, Kouzes and Posner) Take a more expansive view of your circumstances by doing the following (p. 139, Kouzes and Posner):

- “Take the perspective of someone who frustrates or irritates you and consider what that person might have to teach you.”
- “Listen to what other people say: that is, listen to learn rather than to necessarily change their perspective.”
- “Seek out the opinions of people beyond your comfort zone.”

Treat every experience as an adventure. Most of what people do is assigned work. Think of new ways to accomplish tasks.

Leaders experiment and take risks by being willing to do things that have never been done before. It's important to try new things and take chances. "Leaders create the conditions where people want to join with them in their efforts." (p. 147, Kouzes and Posner) Generate small wins by breaking large efforts down into more manageable tasks. Build psychological hardiness by doing the following:

- Stay the course by remaining focused on goals despite setbacks.
- When a suggestion is rejected, determine if it is a desired outcome and come up with ways to convince others of its viability and possibilities.
- "People who experience a high degree of stress, yet cope with it in a positive manner, are psychologically hard." (p. 153, Kouzes and Posner)
- Build psychological hardiness through "commitment, control, and challenge." (p. 153, Kouzes and Posner)
- "With a hardy attitude, you can transform stressful events into positive opportunities for growth and renewal." (p. 153, Kouzes and Posner)

Leaders break things down and accentuate progress by doing the following:

- For large tasks, progress is made incrementally or in an agile way.
- Look for small wins that can be progress toward a larger goal.
- A key to ultimate success is to slowly build momentum in the accomplishment of smaller tasks.
- Small wins build confidence in followers and fill the team with a positive attitude.

Leaders learn from experience. When setbacks happen, take the opportunity to learn from them. Working towards large goals sometimes leads to mistakes along the way. "Overall quality of work improves when people have a chance to fail." (p. 158, Kouzes and Posner) What is important is for people to learn from mistakes to improve the overall outcome. Be an active learner by doing the following:

- There's a strong relationship between how engaged a person is in learning and their leadership capabilities.
- Learning involves "throwing yourself completely into experimenting, reflecting, reading, and getting advice or coaching." (p. 161, Kouzes and Posner)
- Developing a growth mindset is essential to being an active listener.
- People with growth mindsets do not give up on problems despite setbacks.
- Having belief in a positive outcome allows people to make the effort to accomplish goals.

Leaders support their followers' development by creating a climate for learning. They develop resilience to setbacks by picking themselves up and trying again. This demonstrates grit. "Showing grit involves setting goals, being obsessed with an idea or project, maintaining focus, sticking with things that take a long time to complete, overcoming setbacks, and the like." (p. 165, Kouzes and Posner)

Leaders enable others to act by fostering collaboration, creating a climate of trust, and facilitating relationships. People who are unable to trust others fail to become leaders and oftentimes micro-manage others. "When you create a climate of trust, you create an environment that enables people to contribute freely and to innovate." (p. 179, Kouzes and Posner) The leader should be the first to trust. It's a risk to trust others, but this demonstrates self-confidence. Leaders should show concern for others. "The concern you show for others is one of the clearest and most unambiguous signals of your trustworthiness." (p. 182, Kouzes and Posner) Leaders need to share information and knowledge that helps people do their jobs. "One way to demonstrate your competence is to share what you know and encourage others to do the same." (p. 184, Kouzes and Posner)

Leadership is about relationships.

> "Leadership is a relationship that needs to be nurtured and treasured. When leaders can get the people on their team, in their class, or in their community to trust one another, the strength of

> these relationships facilitates the ability of everyone to work together for the collective good. To create an environment in which people know they can count on each other, student leaders need to develop cooperative goals and roles, support norms of reciprocity, structure projects to promote joint efforts, and encourage face-to-face interactions." (p. 186 – 188, Kouzes and Posner)

It's imperative for leaders to develop cooperative goals and roles. "For a team of people to have a positive experience together, they must have shared goals and a clear reason for being together." (p. 188, Kouzes and Posner) Leaders need to trust others and do what they said they would do. "For cooperation to succeed, determine how to design roles so that every person's contributions are both additive and cumulative to the outcome." (p. 189) Leaders support norms of reciprocity. "Cooperation wins over selfishness in the long run. Reciprocity turns out to be a successful approach, because it shows a willingness to be cooperative and an unwillingness to be taken advantage of. Reciprocity leads to predictability and stability in relationships—trust." (p. 191, Kouzes and Posner) Leaders need to structure projects to promote joint efforts. "People are more likely to cooperate if the payoffs for working together are greater than those associated with working by themselves. Cooperative behavior requires people to understand that by working together, they will be able to accomplish something that no one can accomplish individually." (p. 191 – 192, Kouzes and Posner) Leaders need to support face-to-face and durable interactions. "It's the leader's responsibility to provide frequent and lasting opportunities for team members to connect between disciplines, across groups, and among peers." (p. 194, Kouzes and Posner)

Leaders work to strengthen others. Leaders "enable people to take ownership of and responsibility for the group's success by enhancing their competence and their confidence in their abilities, by listening to their ideas and acting on them, by involving them in important decisions, and by acknowledging and giving credit for their contributions." (p. 201, Kouzes and Posner) Leaders can enhance team members' self-determination. "When you feel able to determine your own destiny, when you believe you can mobilize the resources and support necessary to complete a task, then you will persist in your efforts to achieve." (p. 204, Kouzes and Posner) Leaders "realize that leadership actions that increase peoples' sense of self-determination, self-confidence and personal effectiveness make people more powerful and significantly enhance the energy and commitment they put forward." (p. 206, Kouzes and Posner) Leaders need to provide their team members with choices that they can make. "Leaders can reduce followers' sense of powerlessness and accompanying stress that people feel and increase their willingness to exercise their capabilities more fully. You want people to take initiative and be self-directed." (p. 207, Kouzes and Posner) "The only way to create an efficient and effective group of people who can meet the challenges that are part of new things is by giving them the chance to use their best judgment in applying their knowledge and skills." (p. 208, Kouzes and Posner) Leaders must structure tasks to offer latitude to team members. Leaders need to "give people sufficient freedom and choice in deciding how to do their work." (p. 211, Kouzes and Posner) Leaders need to foster accountability among their team members. "When people take personal responsibility and are held accountable for their actions, their colleagues are much more inclined to work with them and are more motivated to cooperate in general." (p. 211). Leaders can create an environment where teammates count on each other to get the job done.

Leaders develop competence and confidence among their followers. Giving team members responsibilities enhances their skills and self-confidence. "Exemplary student leaders strive to create the conditions that make it possible for others to perform effortlessly and expertly despite the difficulty of the task or project." (p. 215, Kouzes and Posner) Leaders educate and share information among their followers. "Developing the competence and confidence of each team member in a group is a virtuous cycle that makes everyone involved feel more qualified, more capable, more effective, and more like leaders themselves." (p. 215, Kouzes and Posner) Leaders organize work to build competence and ownership. “A leader’s job is to enrich the responsibilities of the group members so that they experience variety in their tasks, and opportunities to make meaningful decisions about how things get done.” (p. 217, Kouzes and Posner) Leaders must foster self-confidence. “By building people’s belief in themselves, you are bolstering their inner strength to forge ahead in uncharted terrain, to make tough choices, to face opposition, and the like, because they believe in their skills and decision-making abilities, as well as yours.” (p. 218, Kouzes and Posner) Leadership involves coaching team members. Leaders need to provide constructive feedback to their team members. They express confidence in team members’ abilities to make solid choices. Leaders help team members learn from mistakes and support their decisions. Leaders let their group members “make choices and assume responsibility for them.” (p. 221, Kouzes and Posner) Good leaders ask good questions of their followers. “Asking questions puts leaders in a coaching role, more of a guiding role, which, in turn, frees them up to be thinking more strategically.” (p. 221, Kouzes and Posner)

Leaders encourage the heart by recognizing contributions when they expect the best out of team members and when they personalize recognition. “Exemplary student leaders elicit high performance because they firmly believe in group members’ abilities to achieve even the most challenging goals.” (p. 231, Kouzes and Posner) “You can’t realize the highest level of performance unless you let people know in word and deed that you are confident that they can attain it.” (p. 231, Kouzes and Posner) Leaders bring out the best in people especially when those people did not know it was possible. Leaders “nurture, support, and encourage people they believe in.” (p. 232, Kouzes and Posner) Leaders must show team members that they believe in them. “The expectations you hold as a leader provide the framework into which people fit their realities.” (p. 235, Kouzes and Posner) When you believe that your team members are winners, “you behave in ways that communicate to them that they are precisely that.” (p. 235, Kouzes and Posner) Leaders need to be clear about the goals and the rules. Make sure people know what the expectations are. “Goals and values provide people with a set of standards that concentrates their efforts." (p. 238, Kouzes and Posner) Leaders need to provide feedback and also seek it. Team members' "motivation to perform a task increases only when they have a challenging goal and received feedback on their progress." (p. 239, Kouzes and Posner) "Getting feedback about how they are doing is the only way people know whether they're getting close to their goal and whether they're executing properly." (p. 240, Kouzes and Posner)

Leaders need to personalize recognition, "which is most effective when it comes from and is expressed from your heart." (p. 242, Kouzes and Posner) "You must get close enough to others to know what makes recognition meaningful to them." (p. 242, Kouzes and Posner) Leaders need to get to know people by spending time with them and seeing each team member as an individual. Leaders need to be creative about incentives. "Recognition is most effective when it is highly specific and given close to the time the appropriate behavior occurred." (p. 244 – 245, Kouzes and Posner) "Exemplary leaders make extensive use of intrinsic rewards—rewards built into the task itself, including such factors as a sense of accomplishment, a chance to be creative, and the challenge of the assignment—all directly tied to an individual's effort." (p. 245, Kouzes and Posner) Simply saying "thank you" goes a long way to showing appreciation for team members' efforts. "People naturally feel a little frustrated and unappreciated when someone takes them for granted." (p. 246 – 247, Kouzes and Posner)

Chinese Club

A case study in preparation was the experience of being part of Chinese Club at Bowling Green State University from Spring 1998 through Fall 1999, in which I served as Vice President. Chinese Club was open to anyone, regardless of age, race, or background, with an interest in China, its culture, language, history, etc. Activities included learning to write calligraphy, playing Mah Jong, taking Tae

Kwon Do lessons, and presentations of members' trips to China. Chinese Club also partnered with Japanese Club to co-host organizational meetings.

I helped organize the meetings, determine the budget with the President, learned the Chinese language, assisted our Chinese professor, who was our academic sponsor, and met with other groups' leadership. Chinese Club allowed me to grow my leadership and administration skills in addition to increasing my awareness of cultural values. In the late 1990's and early 2000's, China was becoming a major player in world affairs. So, learning about China and its language were good building blocks for my interest in international business.

Beginning a Career

When students are intentional and strategic about their job search, they are more likely to get the job they want. The most effective means for obtaining internships and jobs is through building relationships with professionals. "All of the job websites in the world, combined, represent only 20% of available jobs, internships, and research positions." (p. 9, O'Keefe) Students who are proactive rather than reactive are more likely to be successful in their job search. Students need to reach out to organizations in more ways than simply sending their resumes. They should make phone calls to companies asking for interviews. They should reach out to companies through LinkedIn. It is about building professional relationships and being persistent.

Resumes that are recommended by coworkers tend to get more attention from Human Resources personnel. If you are interested in organizations, it pays off to spend time researching those organizations, as employers want employees and interns who are genuinely interested in their organizations. "Critical thinking, communication, collaboration, adaptability, creativity, and professionalism will be important skills regardless of your job, function, or industry." (p. 30, O'Keefe) View rejections as learning opportunities. Don't limit your job search only to those opportunities that pertain to your major. The skills that you developed in college can translate to a variety of career paths. Even if a job advertisement appears to require more skills than you currently have in your skillset, it's important to apply for jobs you want, as there may be other jobs available. You can demonstrate to prospective employers that you have the capacity to learn new skills fast. When people tell you that you are not capable of working for a prospective employer or working in a certain field, use that as motivation to prove them wrong by working hard and pursuing your dreams.

Look to professionals for advice and possible mentoring to begin your career or during your career. Job-shadowing is an opportunity for an individual to learn about the day-to-day activities of a particular profession. It's important to do job-shadowing to determine whether the profession is for you or not. Job fairs or career fairs are excellent opportunities to learn about individual jobs, prospective employers, and potential careers. It's not necessary to have your career path figured out in the beginning. That can be figured out along the way when new opportunities present themselves. Many professionals are willing to have conversations and even offer internships to persistent, hard-working, and resilient students. When you graduate from a college or university, take advantage of the alumni network to learn about job opportunities.

It's important to show appreciation to professional connections by saying "Thank you" and being grateful for someone's time. Remember to always be polite, courteous, and respectful toward other professionals. It's important to always prepare for potential job opportunities, interviews, and professional conversations by asking questions and doing research about the people and organizations with whom you are meeting or have an interest. Practicing interview questions is a necessary task before an interview. Practicing coding for technical interviews is helpful in preparation for an interview. Practice listening skills to show attention to detail about the job opportunity and the organization.

When it comes to professional conversations, it's important to have a goal for each one to gain something from them. It's important to follow up with a "Thank you" note whenever another professional offers advice, after a job interview, or other professional correspondence. Doing these builds relationships. Maintaining a network does not require a ton of work, but it does help to occasionally reach out to those people with whom you have communicated or have worked with in the past.

Starting a New Job / Career Transition

When starting a new job, it's important to be curious about the role, the organization, your coworkers' roles, the industry, and the organization's competitors and to ask meaningful questions. Embracing a new role means letting go of what you did in your previous role. When starting a new role, figure out what the new role entails. Sometimes the new role requires learning the breadth of the role more so than managing the details. If the new role is a promotion, "the keys to effective delegation remain much the same: you build a team of competent people whom you trust, you establish goals and metrics to monitor their progress, you translate higher-level goals into specific responsibilities for your direct reports, and you reinforce them through process." (p. 22, Watkins) When a person is promoted, the issues become more complex and ambiguous. Communication is an important aspect when getting promoted. "The people who are the most productive, innovative, and engaged in new roles—the 'fast movers'—are those who establish extremely broad, mutually beneficial, uplifting connections from the start." (Cross, Pryor, and Sylvester)

When starting a new job, focus on business orientation, stakeholder connection, alignment of expectations, and cultural adaptation. To transition to a new role, preparation is required. When starting a new role, it is important to assess your vulnerabilities. It's important to keep an open mind to solving new types of problems. Joining a new company is the equivalent to joining a new culture. It helps to have a strong capacity to learn fast when transitioning to a new role at a new organization. When starting a new role at a new organization, it is important to learn about the organization and the employees working there before implementing new initiatives. Learning about the new role and organization begins with asking questions about how things are done and what happened to get to the current state of operations.

Effective Leadership Traits

Features of an effective leader include the following (p. 9 – 10, Navy):

- Sets the example
- Has learned to be a good follower
- Knows his or her job
- Establishes objectives and plans for their accomplishment
- Knows himself or herself and seeks self-improvement
- Takes responsibility for his or her actions, regardless of their outcome
- Is consistent but not inflexible
- Seek responsibility and develops a sense of responsibility among his or her subordinates
- Treats every person as an individual, not as a number
- Keeps his or her subordinates informed
- Encourages subordinates to offer suggestions and/or constructive criticism
- Makes sure the task is understood, supervised, and accomplished
- Trains his or her subordinates as a team

- Employs his or her unit in accordance with its capabilities

Traits of effective leaders also include the following (p. 38 – 40, Lussier and Achua):

- Dominance
    - To be an effective leader, an individual must want to be a leader.
- High energy
    - “Leaders have high energy with a positive drive to work hard to achieve goals. They focus on the positive and have stamina and tolerate stress well.”
- Self-confidence
    - “Leaders display self-assurance about their abilities and foster confidence among followers.”
- Locus of control
    - “Leaders take responsibility for who they are, for their behavior and performance, and for performance of their organizational unit.”
- Stability
    - “Stable leaders are emotionally in control of themselves. They don’t let their anger have negative outcomes.”
- Integrity
    - “To be viewed as trustworthy, leaders need to be honest, support their followers, and keep confidences.”
- Intelligence
    - “The manager’s job calls for a high degree of intelligence, and leaders generally have above-average intelligence.”
- Emotional Intelligence

    - "Emotional intelligence is the ability to work well with people, and it is essential to healthy relationships."
    - Emotional intelligence is composed of self-awareness, social awareness, self-management, and relationship management.
- Flexibility
    - "Effective leaders are flexible and adapt to the situation."
- Sensitivity to others
    - "Sensitivity to others refers to understanding group members as individuals, what their positions on issues are, and how best to communicate with and influence them."

Major leadership traits include the following (p. 10, Navy):

- Integrity
- Dependability
- Cooperation
- Loyalty
- Unselfishness
- Sense of humor
- Tact
- Ability to write well
- Ability to speak effectively
- Initiative
- Judgment
- Enthusiasm
- Creativity

- Decisiveness
- Endurance
- Self-discipline
- Moral and physical courage

What to Know When Becoming a Leader

Managers are not necessarily leaders. "The ability to lead is really a collection of skills, nearly all of which can be learned and improved. Successful leaders are learners and the learning process is ongoing. A result of self-discipline and perseverance." (p. 13 - 14, Maxwell, Leadership 101) Becoming a leader is a daily routine, and it is hard work. The key to success is self-discipline, through daily work habits. Leaders must avoid making excuses and accept responsibility. Rewards should be given when the work is complete. Leaders remain focused on results and the process to get there. Establishing goals and working hard to achieve them are essential to success. Leaders must be able to juggle high priority projects. Leaders must determine what they must do and what they can delegate.

"Trust is the foundation of leadership." (p. 45, Maxwell, Leadership 101) A leader must have competence and character, and he or she must be able to make connections with others. "Character makes trust possible and trust makes leadership possible." (p. 47, Maxwell, Leadership 101) Leaders should make strong character a high priority, because it communicates much about them. Part of demonstrating character is to show that you have integrity.

It's essential for a leader to have vision to show their followers where the team or organization is going. For leaders to have vision, it's important for them to have mentors to help them see far ahead.

It's important for leaders to have the right attitude to get people to follow them. Leaders must empower their followers to lead. Leaders pass their knowledge on to their followers.

If an individual wants to make an impact, they must increase their influence. True leadership comes from influence. Leaders need to influence others to lead in a new direction. Leaders need to influence by getting followers to participate. "A leader's responsibility is to develop others to do the work." (p. 79, Maxwell, Leadership 101) Leaders need to help followers grow personally.

Chapter 2: Personal Professional Development

Motivation

"Motivation is what gets us to act or do something and is clearly linked to personality. Motivation is about setting goals and going out to meet them; our personality strongly influences what these goals or drives might be." (p. 148, Mann) "Motivation is anything that affects behavior in pursuing a certain outcome. Motivation is a quest for personal gain. As a leader, your job is to motivate your followers to work hard." (p. 79, Lussier and Achua) Motivation is driven by psychological needs. "Motivation is what translates learning into observable behavior." (p. 338, Houston, et al.) Motivation is made up of drives and incentives, which is influenced by the surrounding environment. "Human behavior is flexible and responsive to variations in the environment." (p. 341, Houston, et al.) The need to achieve or exceed standards of excellence motivates people. People are motivated by fear of failure and hope of success. People are motivated to work when they feel comfortable and work within an environment that encourages them to engage in training and self-improvement, while they feel like they are working with a purpose. People should be motivated to continue to learn new skills throughout their careers.

Software engineers should be motivated to solve problems by developing elegant technical solutions. They should want to create something out of nothing, or rather, take an idea and build a system that accomplishes a goal. Software engineers are problem-solvers and builders. The desire to build a functioning system is inherent in the human psyche. It goes beyond financial rewards and satisfies a craving that people must fulfill the need of creativity. System development requires vision to accomplish ideas that people have. Software engineers should take it upon themselves to engage in training, even when their employers do not support their training efforts. When software engineers are motivated, they find new functionality to implement in software, and they feel encouraged to engage in self-learning of technologies and methodologies. Software engineers should continuously engage in self-training throughout their careers, as technology constantly changes.

Leaders "can establish an environment that allows motivation to flourish, which facilitates this trait in their subordinates." (p. 152, Hess) "Leaders must be willing to tackle complex organizational problems, exert their influence, and be committed to the social good of the organization." (p. 61 – 62, Northouse) Leaders who want their teams to succeed understand what motivates the team members. Leaders should provide assignments to team members that expand their knowledge and abilities. Assignments should be challenging and motivating to team members. They should allow people to grow their capabilities beyond what they can currently do. Leaders may want to encourage "citizenship behaviors". Having a reason to show up to work every day motivates people to give their best efforts. If a manager is not motivating an employee, the employee owes it to himself or herself and their career to find ways to motivate themselves. "It is important that managers help employees find the best possible ways to go beyond the call of duty in order to help make work more meaningful and less depleting. Leaders may want to practice 'citizenship crafting', where people redesign their work by altering aspects of the job itself, the people with whom they work, and their mindset about their jobs in ways that play to their strengths, motives, and passions." (Bolino and Klotz)

"There is no stronger motivation for employees than understanding that their work matters and is relevant to someone or something other than a financial statement. Motivation is less about employees doing great work and more about employees feeling great about their work. The better employees feel about their work, the more motivated they remain over time." (Lai) Employees will appreciate it when their leaders share with them the context of the work – what the organization is doing, why, and who benefits. "Employees are motivated when they can make progress without unnecessary interruption and undue burdens." (Lai) "Most productivity studies have found that motivation has a stronger influence on productivity than any other factor." (p. 138, McConnell, Professional Software Development) The main reasons people work are play, purpose, emotional pressure, economic pressure, and inertia. “Culture is the set of processes in an organization that affects the total motivation of its people.” (McGregor and Doshi) When leaders explain why their team is working, this purpose is motivational to the team members.

“Almost every organization depends on human labor to fulfill its reason for existing. Yet, the leaders of most organizations spend very little time ensuring their workers receive effective motivational supervision. The fact is that supervisors simply being smarter or working harder themselves can never equal the added value that can be achieved by inspiring every member of a work team to give their best efforts every day.” (p. 15 – 17, Tower) Supervisors should customize their coaching process to fit individuals. To motivate others, a leader must be empathetic. Supervisors should acknowledge good work being done by praising the employee with their name. Understanding human insecurity can aid you in becoming a motivational leader. “Maximally effective management is the art of getting the right things done through others—willingly.” (p. 33, Tower)

Abraham Maslow – Maslow’s Hierarchy of Needs

“Abraham Maslow believed that needs are physiological or psychological deficiencies that a person feels a compulsion to satisfy. These needs create tensions that can influence a person’s attitudes and behaviors. Maslow’s theory is based on the following two principles:” (p. 149, Mann)

- The deficit principle: “This states that a satisfied need no longer motivates behavior.”
- The progression principle: “Maslow identified five needs that he believed were arranged in a hierarchy; the next level only becomes relevant when the previous level has been satisfied.”

Figure 1 shows Maslow's Hierarchy of Needs. Biological needs must be met first. Safety and security needs must be met second. Belongingness and love needs must be met thirdly. Self-esteem needs must be met fourth. Self-actualization is then achieved. A person who reaches self-actualization "is spontaneous, creative, and has a sense of humor. Such a person is in tune with the meaning and mystery of life. Lower needs do not have to be totally fulfilled before we move upward. Lower needs must be relatively satisfied; they must not be so strong and pressing that they block our higher efforts." (p. 346, Houston, et al.)

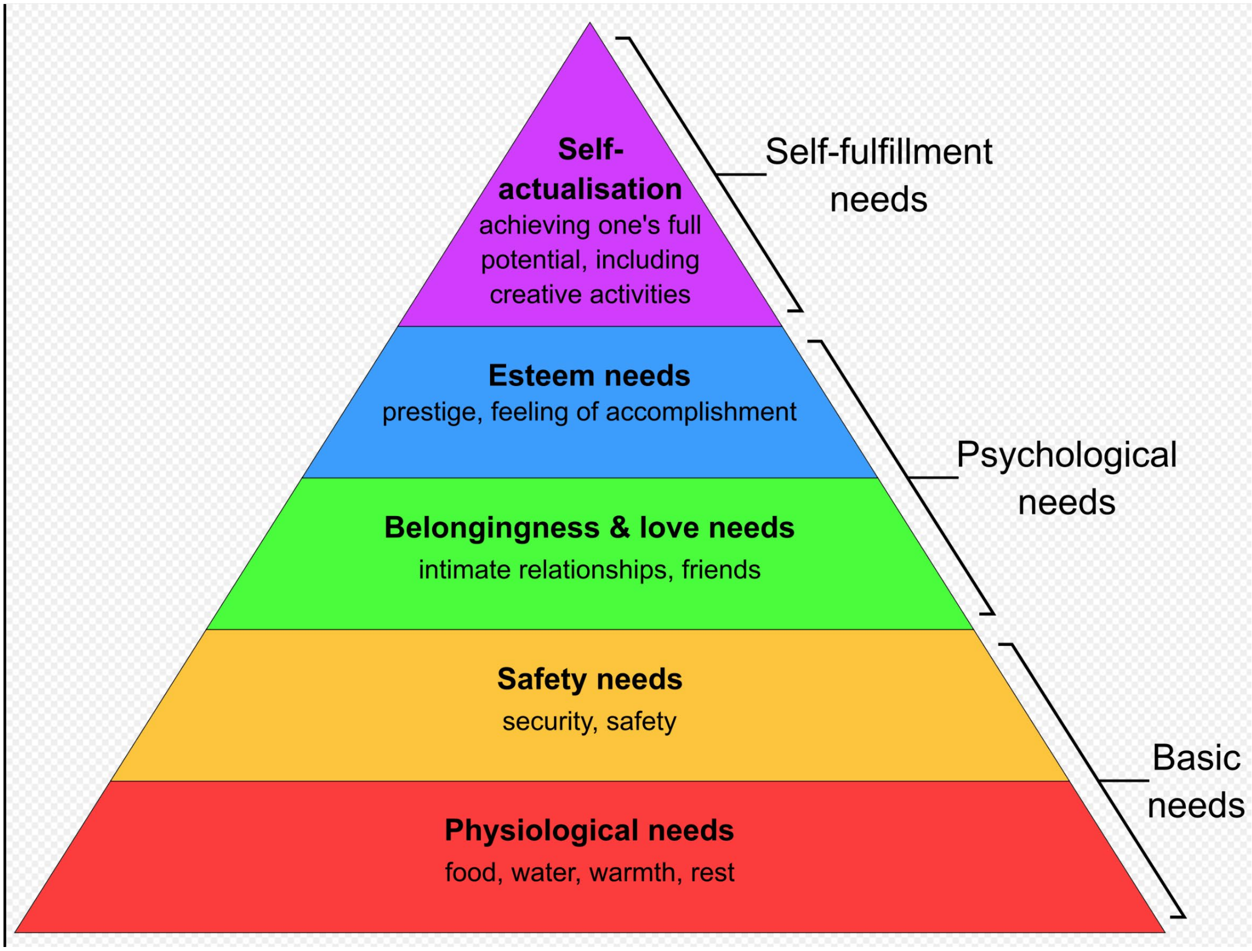
Self-actualisation
achieving one's full potential, including creative activities
Self-fulfillment needs
Esteem needs
prestige, feeling of accomplishment
Belongingness & love needs
intimate relationships, friends
Psychological needs
Safety needs
security, safety
Physiological needs
food, water, warmth, rest
Basic needs

Figure 1 Maslow's Hierarchy of Needs (Source: Wikipedia)

Software engineers must satisfy their needs on the Hierarchy just like any other human must do. Developers are motivated by the same factors that motivate people of all walks of life. While developers are climbing the Hierarchy, they can learn and practice software development skills. Hopefully, they can reach the level of self-actualization and focus their efforts on building outstanding software solutions while stability provides them with the support they require. “What self-actualization means is that a person is able to recognize that they have potential and as such, pursue that highest potential. A self-actualized individual is someone who believes in becoming the very best that they can and pursues it to all ends. The self-actualized person seeks greatness for themselves.” (p. 7, Smith)

Motivating Others

Power

“Power can be good when wielded with respect. When it unites people for a shared purpose and motivates them to be the best version of themselves.” (p. 3, Rometty) Power means exercising influence to get people to change. “The concept of power is related to leadership because it is part of the influence process. Power is the capacity or potential to influence.” (p. 10, Northouse) Power and leadership used to be synonymous. “Followers must consent to the governing influence of managers for the organization to be successful. Today’s effective leaders are relying less on position power and more on personal power to influence others, and they are open to being influenced by followers with personal power.” (p. 110, Lussier and Achua) Power can cause people to lead in destructive ways or beneficial ways. To manage power effectively and be an influential leader, it is important to embrace authenticity and to be honest with yourself and others. Power comes in the form of recognizing opportunities and seizing on them. Having power starts with an honest self-assessment of who you are. There's power in having goals and breaking those goals down into more manageable goals to incrementally accomplish. Power comes from working hard and diligently and creating a reputation for yourself. Be willing to help others, as that builds your credibility and power. True power requires sharing your time with others. Helping others is key to demonstrating your power. People holding power should listen to be of benefit to others. It's important to be an effective and positive force. There's power in joining forces with others to solve problems.

The Six Bases of Power are (p. 11, Northouse):

- Referent power: "Based on followers' identification and liking for the leader. A teacher who is adored by students."
- Expert power: "Based on followers' perceptions of the leader's competence. A tour guide."
- Legitimate power: "Associated with having status or formal job authority. A judge."
- Reward power: "Derived from having the capacity to provide rewards to others. A supervisor."
- Coercive power: "Derived from having the capacity to penalize or punish others. A coach."
- Information power: "Derived from possessing knowledge that other want or need. A boss."

Referent power is "based on the user's personal relationships with others. It's about appealing to the follower's values, ideals, and aspirations. It appeals to emotions. It's about understanding the values, hopes, fears, and goals of followers. Be positive and optimistic and create a vision of how things will be when the objective is achieved." (p. 115, Lussier and Achua)

Society is completely dependent on computers and software, which gives programmers power and requires them to be responsible and not cause harm. Software engineers have the power and responsibility for creating applications that disrupt the status quo. Software engineers must have long-term context in their decision-making. They must protect privacy and security. Holding a great deal of power means that software engineers must be ethical. Software engineers must be responsible, as society now depends greatly on the software systems that they produce.

Professional Skill Development – Pride in Craft

"Craftsmanship is the state of knowing how to do something well and is the outcome of good tutelage and lots of experience." (p. 9, Martin, Clean Craftsmanship) "Craftsmanship is the desire to do a job well for its own sake. Craftsmanship focuses on objective standards, on the thing itself. The good craftsman, moreover, uses solutions to uncover new territory; problem solving and problem finding are intimately related to his or her mind." (p. 9 – 11, Sennett) A craftsman is "a person who is deeply skilled and accomplished in a particular activity—someone who is comfortable with their tools and their trade, who takes pride in their work, and who can be trusted to behave with the dignity and professionalism of their calling." (p. xxii, Martin, Clean Craftsmanship)

"A person needs to know about 50,000 chunks of information to be an expert in a field, where a chunk is any piece of knowledge that can be remembered rather than derived." (p. 37, McConnell, Professional Software Development) About 10,000 hours of experience are required to become a master craftsman. Programmers are craftsmen. They fix bugs and "see new possibilities open up for the use of the code. The code is constantly evolving, not a finished and fixed object." (p. 26, Sennett) What matters for a craftsman is what he or she contributes to the discussion, or the ongoing craft with which they are working. It's the quality of the contribution or the work that matters. Craftsmanship is about developing a skillset that depends on repetition and practice. For the craftsman, the hand and the mind work together to implement a design. The craftsman treats his or her workshop as their home, and it is important for software engineers to set up their workstations or offices in a way that is ergonomic, productive, and conducive to completing programming tasks. It is normal to use dual monitors with a docking station at workstations. Likewise, it is important to use a comfortable, yet ergonomic chair.

Software engineering requires discipline, ethics, and standards. Software engineers are craftsmen, just as much as stonemasons, blacksmiths, carpenters, and engineers are craftsmen. Software development is a trade that requires skill that is learned and developed over time. Aspects of programming and coding can be learned in relatively short periods of time, but the process of software engineering takes longer to master. Software engineers must take pride in their craft and continuously hone it if they want to become expert at it. Like the craftsman of years ago, the software engineer will oftentimes work for different employers early in his or her career, gaining experience and knowledge. Gaining experience from different jobs may require the software engineer to move to different cities, similarly to the journeyman ways of gaining experience and knowledge. Depending on the employer, the circumstances, the project, and more, the software engineer may decide to settle down for a while in a location before moving on to new employers and projects. As the software engineer encounters opportunities and completes project over the years, he or she becomes an authority on software development.

When software engineers are responsible and take their craft seriously, they can work well within the framework of a team and an organization. "A key to performing well is accountability:

making commitment, working in ways you are proud of, and rendering account of my activities clearly and directly. Working well and visibly builds strong relationships. Offer accountability as a way of demonstrating trustworthiness and encouraging your own best behavior. Accountability can be offered, asked, even demanded, but it cannot be forced." (Beck) Continuous integration requires development team members to be accountable to one another.

The software engineer, as a craftsman, must adhere to principles of programming methodologies, such as object-oriented programming and design patterns, to produce systems that reflect industry standards. Following industry standards means that software engineers put integrity in their designs. Software engineers must be detail-oriented and be aware of every part of the code they create down to every character that they type.

Professional Software Development

Whether they are called programmers, software developers, software engineers, or other job titles, the work of software development requires a level of professionalism that can only be attained through education, training, and practical experience. Programmers do not just want to write code; they want to design great software. Software engineers have a great desire to build things. To be a software engineer requires total and absolute commitment to completing projects on time and within budget. The typical software engineer is dedicated to completing a project rather than quitting a job in the middle of a project.

Software development projects should be conducted with an engineering approach, mindset, and process. The process is often called the Software Development Lifecycle (SDLC). Requirements need to be gathered, followed by analysis and design, programming and testing (or quality assurance), delivery, and maintenance. Planning and designing software should happen before programming it. When programmers jump into coding without a full awareness of the requirements, problems and delays in the project are bound to transpire. Likewise, testing of software should happen before it is delivered to ensure the customer receives a high-quality solution. Indeed, test design and tests themselves should be written before programming, or construction of the software commences. The reason for understanding requirements, planning and designing the system, and preparing testing before programming is that software is complex, and humans are prone to error, resulting in bugs in the software. "Several studies have found that 40 to 80 percent of a typical software project's budget goes into fixing defects that were created earlier on the same project." (p. 12, McConnell, Professional Software Development)

Ongoing Development, Learning, and Training

For a software engineer, there is no substitute for practical experience. Writing code, reading code, and studying other people's code are essential to becoming a better programmer. Books and training videos abound to help software developers improve their craft. Training sites include YouTube, Pluralsight, Udemy, LinkedIn Learning, and many more. It's standard to post a career profile on LinkedIn and regularly update it. It's also important for software engineers to continuously develop their skillsets, keep up on the latest technologies, and pursue further education and training. Developers can post code examples on GitHub and develop websites that act as portfolios that demonstrate their skillsets. In addition, software engineers can decide to obtain additional education, such as a master's degree or a post-graduate degree. If a software engineer has a bachelor's degree in computer science, it may make sense to broaden their horizons by pursuing a business or science degree. Hobbies can enhance skillsets as well, if a developer wants to learn about subjects such as electrical engineering or robotics. He/she can join a local group that specializes in technical pursuits. Various cities around the United States offer Makerspaces where people can work on their side projects. Meetup groups offer gatherings for improving knowledge and networking in a wide variety of areas. Professional organizations exist to allow developers to network with other Information Technology (IT) or engineering professionals, such as the Association of Computing Machinery (ACM) and the IEEE Computer Society. Volunteer work is a way for developers to gain additional skills and to build their professional profiles, while also networking. Tutoring students in STEM (Science, Technology, Engineering, and Mathematics) provides software engineers with an outlet to develop teaching skills and learn to be mentors for their careers. Reading and writing about technical topics are opportunities to explain concepts to others and helps software engineers learn to explain subject matter to others. Answering questions on websites, such as Stack Overflow, demonstrates your expertise and boosts your online profile.

Chapter 3: Habits and Productivity

Habits

Building good habits is essential in personal and professional lives and they are essential to accomplishing goals.

Productivity

Motivated individuals and organizations strive to be as productive as possible. Productivity is measured by output and sustained productivity is a goal of motivated individuals and organizations.

Chapter 4: Problem Solving

Asking Questions and Developing Ideas

Problem solving involves asking questions and developing ideas. Leaders and their teams should be critical of their ideas to determine if they are viable. “Leaders are not responsible for coming up with all ideas. Rather, they need to provide the environment for everyone to come up with ideas.” (Sinek) The engineering team should encourage its members to continue to have ideas. “Questioning is a uniquely powerful tool for unlocking value in organizations: It spurs learning and the exchange of ideas, it fuels innovation and performance improvement, it builds rapport and trust among team members. And it can mitigate business risk by uncovering unforeseen pitfalls and hazards. Asking a lot of questions unlocks learning and improves interpersonal bonding. It’s important to ask open-ended questions, as they can be particularly useful in uncovering information or learning something new.” (Brooks and John)

> “Thoughtful and well-formulated questions frequently lead to solutions to difficult and perhaps initially intractable technical problems. Leaders must engage many people, especially team members, to take advantage of their collective wisdom and experience, and thereby gain the broadest and most comprehensive outlook possible. When a questioning culture exists, change is facilitated because team members are continually probing and evaluating the status quo and so are open to reconsideration and re-evaluation of decisions and conclusions. Questions should be asked even when people assume that questions do not need to be asked. It’s especially important to ask questions when people perform tasks because that’s the way the things have always been done. Questions should be open-ended to incite discussions that reveal new information, directions or insight, present the reasoning behind opinions and conclusions reached, and identify both subordinate and leader biases.” (p. 118 – 124, Hess)

People need to admit that they don't have all the answers, and they should ask questions to find the answers. "Asking for help is a strong signal to others that you are trusting, and you're more likely to be trusted in return. The kind of questions leaders need to ask are those that invite people to come together to explore major new opportunities that your organization hasn't identified yet." (Hagel) Asking questions can involve a concept called "question burst – which is the process of asking enough questions at a rapid pace to become uncomfortable enough to generate fresh ideas. People usually experience one of three outcomes after participating in a question burst: a more positive state of mind about their challenge that leads to new, valuable ideas; a feeling that the problem is much bigger than they expected; or a realization that they are part of the problem." (Somers)

Teams must ask why problems occur and then proceed with asking how problems can be resolved. The same concept applies to new ideas – ask why something must be accomplished, and then, ask how it will be accomplished. Asking "how" a problem will be solved is sometimes faster at addressing the issue than asking "why" it needs to be solved. Asking "why" a problem needs to be solved is often the job of the business side of an organization, while asking "how" a problem will be solved is often the job of the engineering side of an organization. When obstacles are encountered, the "why" of it happening must be overcome with the "how" of what to do to get past the obstacle. Asking "why" is backwards-focused, whereas asking "how" is focused on the future. "To ask the right question is already half the solution to a problem." (Carl Jung) Visual imagery is a useful method of thinking to generate ideas to solve problems. Problem solving can become a habit, as people encounter more and more problems. It's typically better to come up with alternative solutions to problems rather than quickly settling on the first answer.

When attempting to solve problems, people often encounter perceptual, emotional, or cultural blocks. "Perceptual blocks are obstacles that prevent the problem-solver from clearly perceiving either the problem itself or the information needed to solve the problem." (p. 15, Adams) Perceptual blocks can involve stereotyping, such as premature labeling, difficulty in isolating the problem, tendency to delimit the problem area poorly, or failure to utilize all sensory inputs. Problems can be obscured by inadequate clues or misleading information. "In engineering, people occasionally become so involved in attempting to optimize a particular device that they lose sight of alternate ways to alleviate the difficulty." (p. 24, Adams) Inadequate problem definition is one of the challenges of problem solving. "A relatively small time spent in carefully isolating and defining the problem can be extremely valuable both in illuminating possible simple solutions and in ensuring that a great deal of effort is not spent only to find that the difficulties still exist." (p. 25, Adams) One of the emotional blocks that can prevent the generation of ideas is the inability to tolerate ambiguity. Additionally, judging ideas rather than generating them is another emotional block. "One of the better ways of overcoming an emotional block is to realistically assess the possible negative consequences of an idea." (p. 46, Adams) The mind should be allowed to struggle with problems over time. "Excessive motivation to succeed, especially to succeed quickly, can inhibit the creative process. The creative person must be able to vividly form complete images, but also to manipulate them. When people are at their most creative, they are in a state called flow in which they are completely consumed by their task." (p. 51 – 54, Adams) These types of blocks can be busted by the following techniques:

- Having a questioning attitude
- Brainstorming
- List making
- Using other people's ideas

Inventors and innovators come up with great ideas, and they need individuals, or teams of individuals, who can help translate those ideas into tangible products and systems. Delivering on a great idea requires the efforts of more than one person. “The biggest challenge to deliver anything on time or within budget will always be the lack of clarity in either direction or requirement.” (p. 61, Taveau)

Problem Solving

“Defining problems in painstaking detail is difficult work.” (p. 5, McConnell, Professional Software Development) “Problem-solving is a mental process that involves discovering, analyzing, and solving difficulties or conundrums. The ultimate goals of problem-solving is to overcome obstacles and find a solution that best resolves the issue.” (p. 93, Mann) “Problem-solving happens when people, in the spirit of bridging differences, embrace tensions that arise from opposing forces.” (p. 3, Rometty) “Problem solving is decision making when there is complexity and uncertainty that rules out obvious answers, and where there are consequences that make the work to get good answers worth it.” (p. xiii, Conn and McLean) Skepticism is a healthy attitude to have when it comes to problem-solving. Skepticism makes a person ask, “What is the evidence?” and “What are the facts?” Problem-solving should be based on factual information rather than opinions or hearsay. It’s necessary to maintain objectivity when solving problems. “The person with good intellectual health suspects all varieties of ready-made answers, seeks evidence, recognizes evidence when it appears or is found, and strives to ensure objectivity in observations and interpretations.” (p. 18, Navy) The first goal of problem solving is to identify the problem. One of the requirements for problem solving is to have curiosity. When solving problems, learning from failures is essential. An important part of problem solving is to ensure that the team members have the capabilities required to solve those problems.

Conn and McLean offer the following steps necessary to solving problems:

- Define the problem

- Disaggregate the issues
- Prioritize the issues
- Build a work plan and timetable
- Conduct critical analyses
- Synthesize findings from the analysis
- Prepare a powerful communication

Problem solving involves “using a hypothesis to bring forth the arguments to either disprove it or support it. Constraining the scope of the problem with clear boundaries makes problem solving more accurate and speedy. A logic tree helps people see the structure of a problem in one picture and helps people break up the analyses into manageable chunks.” (p. 13 - 14, Conn and McLean) See logic tree diagram in Figure 2.

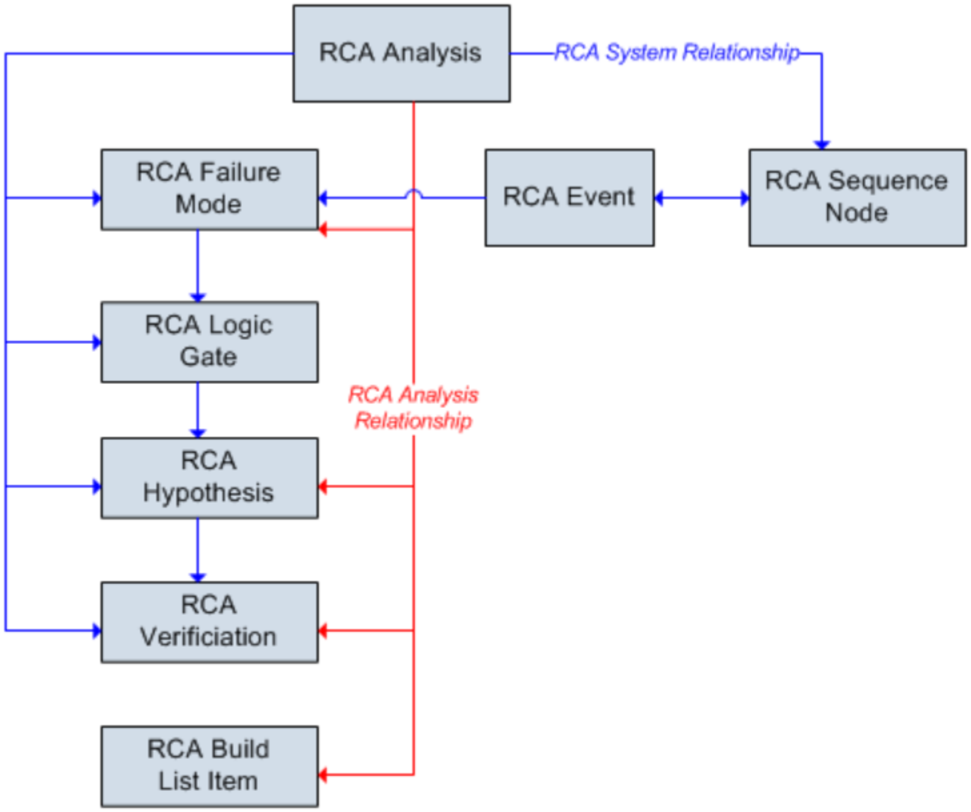

Figure 2 Logic Tree (Source: General Electric)

Problem solving involves root cause analysis, which involves getting to the heart of a problem. Toyota makes great use of root cause analysis to solve problems. See Root Cause Analysis diagram in Figure 3.

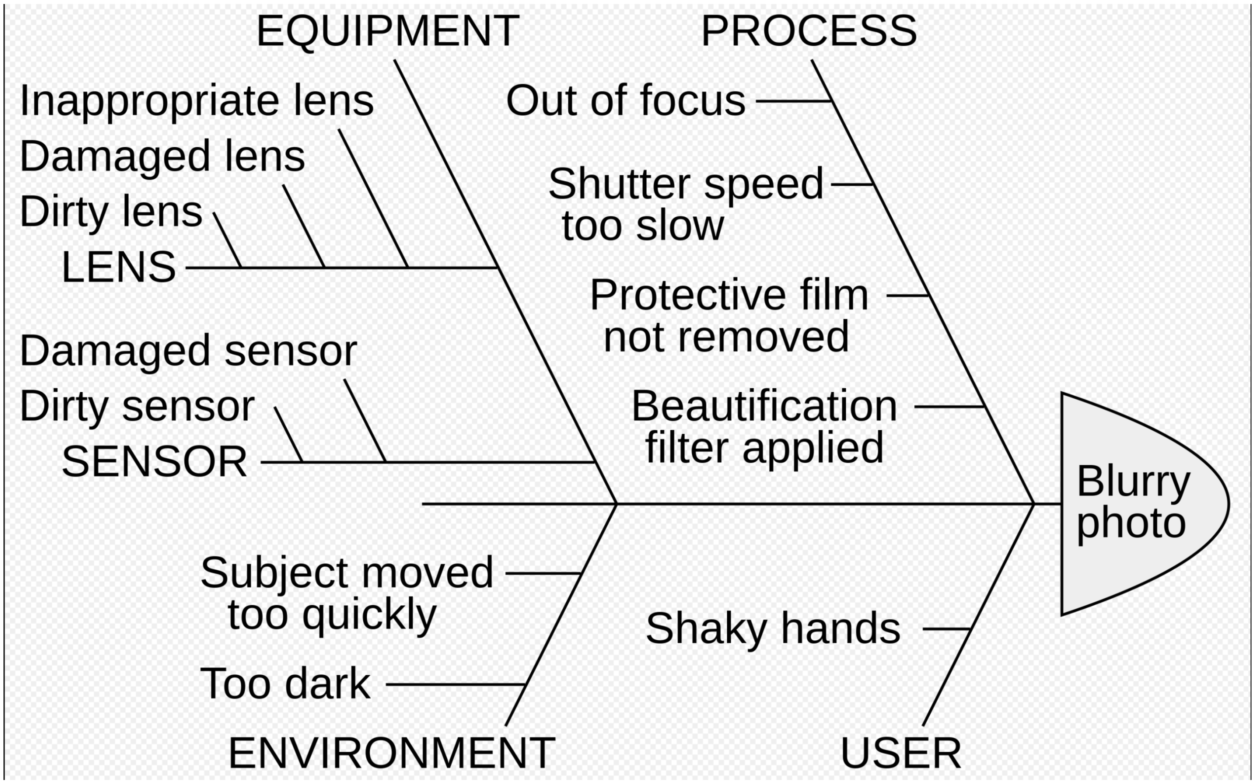

Figure 3 Root Cause Analysis (Source: Wikipedia)

There are issues that prevent problem-solving, such as the following (p. 93 – 94, Mann):

- Functional fixedness
  - Cognitive bias “limits a person to using an object in the way it is traditionally used.”
  - It “comes into play when an individual is unable to see all the options available to solve a problem.”
- Mental set
  - This “refers to an individual wanting to immediately use a problem-solving strategy that they used in the past to solve a problem.”

  - Other options are ignored.
  - It “prevents people from thinking up new ways of solving a problem.”

Other issues may prohibit problem solving. Those issues could be political or cultural, depending on the working relationships of the coworkers. Additionally, lack of communication can be a hindrance to the problem solving process.

Software engineering is about problem solving that is done with the use of computers and digital devices. Software was invented to solve problems. Problem solving in the realm of software engineering involves the following steps:

- Problem is identified
- Investigation into why the problem is happening, which involves root cause analysis
- Solutions are determined
- A solution to the problem is selected
- The solution is implemented and tested in the software
- The solution is deployed
- The solution is monitored

When a team is involved in problem solving, they usually employ brainstorming to generate ideas for why a problem happens as well as potential solutions to the problem. Brainstorming includes listing out pros and cons of potential solutions. Depending on the type of problem solver a person is, they have both strengths and impediments to problem solving. People from different backgrounds bring different perspectives to problem-solving because they think differently. People’s problem-solving abilities can be enhanced by adopting the perspectives of different people. One of the requirements for problem solving is to assemble a team of people who think differently and like to solve problems. A team that collaborates to solve problems must share knowledge with each other to be effective. Sharing knowledge with each other makes the team more cohesive.

Binder and Watkins recommend that before solving a problem, efforts can be helped by doing "problem-framing, which is a process for understanding and defining a problem. Problems should be examined from all angles and their complexities explored. Effective problem-framing lets you uncover new insights and generate fresh ideas." (Binder and Watkins) The problem-framing approach involves the following: expand, examine, empathize, elevate, and envision. Binder and Watkins:

- Expand: "A comprehensive exploration of an issue and its nuances"
- Examine: "A deep dive to identify root causes of the problem"
- Empathize: "Focus on the stakeholders—employees, customers, clients, investors, etc."
- Elevate: "Explore how the problem connects to broader organizational issues"
- Envision: "actively design solutions to the problem"

Oftentimes with problem solving, failure happens because the problem is poorly defined. According to Conn and McLean, good problem statements involve the following (p. 33, Conn and McLean):

- They focus on outcomes.
- They are specific and measurable.
- They adhere to time constraints.
- They are "designed to explicitly address decision-maker values and boundaries, including the accuracy needed and the scale of aspirations."
- They are "structured to allow sufficient scope for creativity and unexpected results."
- They are "solved at the highest level possible."

When determining the problem statement, it can help to redefine the problem at a higher level. Problem statements are better when they are based on facts. Relating problem statements to user experience is sound practice. In software engineering, user experience is one of the goals of designing software.

Brainstorming is an effective method for both determining why a problem is happening and how a solution will be developed. “Brainstorming is a process to generate creative ideas through group discussion.” (p. 48, Agarwal) Criticism of ideas is put on hold. According to Agarwal, brainstorming is facilitated by doing the following (p. 49 – 52, Agarwal):

- Prepare the environment
- Provide an overview of the problem
- Set the schedule and establish the rules
- Run the session
- Organize ideas and act

With problem solving, “it is critical to know what is off limits and to understand the implications of the boundaries to the solution set.” (p. 37, Conn and McLean) When solving problems, it helps to break those problems down into smaller, more manageable constituent problems. Before analyzing a problem, determine what the outcome or goal is. During problem solving, it’s important to determine what the variables are during analysis. A good problem solving team requires a great team leader. Agile project management is being used commonly to solve problems. Successful problem solving involves a strong amount of analysis to determine both what the problem is and what solutions may be possible.

“A leader’s toolset involves the creative ability to solve new and unusual ill-defined problems. Skills include being able to define significant problems, gather problem information, formulate new understandings about the problem, and generate prototype plans for problem solutions.” (p. 62, Northouse) "Some leaders are finding roles as creative problem solvers. Depending on the problem, these leaders either step up to lead the solution of a problem, or they step aside to let another person lead the effort. This type of leader leans on their expertise to solve a problem. Problem-solving skills of leaders include the following (p. 62 – 63, Northouse):

- Problem definition
    - "The ability to define noteworthy issues or significant problems affecting the organization"
- Cause/goal analysis
    - "The ability to analyze the causes and goals relevant to addressing problems"
- Constraint analysis
    - "The ability to identify the constraints, or limiting factors, influencing any problem solution"
- Planning
    - "The ability to formulate plans, mental stimulations, and actions arising from cause/goal and constraint analysis"
- Forecasting
    - "The ability to anticipate the implications of executing the plans"
- Creative thinking
    - "The ability to develop alternative approaches and new ideas for addressing potential pitfalls of a plan identified in forecasting"
- Idea evaluation
    - "The ability to evaluate these alternative approaches' viability in executing the plan"
- Wisdom

  - "The ability to evaluate the appropriateness of these alternative approaches within the content, or setting, in which the leader acts"
- Sensemaking/visioning
  - "The ability to articulate a vision that will help followers understand, make sense of, and act on the problem"

Leaders need models to help explain problems and how to solve them. "Organic models lead toward problem solving in which everyone benefits." (p. 11, Weinberg) Systems thinking depends on the fundamental idea of underlying organic models. Leadership depends on the organic model, in which people are empowered in the contained environment, which contains these aspects: motivation, organization, and ideas or innovation. "All of the most consistently successful technical leaders empower people by the value they place on innovation, on doing things in a better way. They concentrate on three major areas: understanding the problem, managing the flow of ideas, and maintaining quality." (p. 21, Weinberg)

Understanding the problem involves the following steps: (p. 27 – 28, Weinberg)

- "Read the specifications very carefully."
- "Encourage teammates to read the specifications very carefully."
- "Resolve arguments by referring back to the original problem."
- "Seek clarifications and additional information about the specifications from the customer."
- "Refer back to the specifications after work has proceeded for a while, when the implications of some of the requirements can be better understood."

Managing the flow of ideas is composed of the following aspects: (p. 29 – 31, Weinberg)

- "Contribute a clever idea to the team."

- “Encourage copying of useful ideas.”
- “Elaborate on an idea that a teammate contributed.”
- “Drop one’s own idea in favor of an idea the team wants to develop and refuse to let an idea drop until everyone understands it.”
- “Resist time pressure and take the time to listen when other people explain their ideas.”
- “Test ideas contributed by other people.”
- “Withhold quick criticism of teammates’ ideas, in order to keep the ideas flowing.”
- “When you must criticize an idea, make clear that you are criticizing the idea, not the person who offered the idea.”
- “Test your own ideas before offering them.”
- “When time and labor are running short, stop working on new ideas and just pitch in.”
- “Encourage the team to drop ideas that had succeeded earlier but cannot be extended to the new situation.”
- “Revise a dropped idea later, when it has value for another part of the problem.”

Maintaining the quality involves the following steps: (p. 32, Weinberg)

- “Measure quality as the project proceeds.”
- “Design tools and processes to measure quality as you build a solution.”
- “Measure the speed of implementation, compare it to the schedule, and be prepared to change the solution procedure.”
- “Step back from the project to refresh your perspective and to assess its viability.”
- “Check ideas with the customer before implementing them.”
- “Restore morale when an idea collapses.”

Successful technical leaders face obstacles to motivating others. They need to focus on the interaction between people when they communicate with each other. Leaders also need to either put the task ahead of the people or put the people ahead of the task.

> "When survival is concerned, there's no choice but to put the people first. If the job isn't highly technical, the leader need not be competent, but can lead by fear. People with strong technical backgrounds can convert any task into a technical task, thus avoiding work that they don't want to do. Leaders who don't care about people don't have anyone to lead unless their followers don't have a choice. No amount of caring for people will hold your audience if you have nothing to offer but pretend you do. Task-oriented leaders tend to overestimate their accomplishments. Very little work we do is so important that it justifies sacrificing the future possibilities of the people doing the work. In a complex environment, even the most task-oriented leader is forced to put people first, or the task won't get done. To be a successful problem-solving leader, you must keep everyone's humanness at the forefront. If you are a leader, the people are your work. There is no other work worth doing." (p. 122 – 127, Weinberg)

There are some lessons that leaders should be aware of when helping others:

> "Wanting to help people may be a noble motive but that doesn't make it any easier. If people don't want your help, you'll never succeed in helping them, no matter how smart or wonderful you are. Effective help can only start with mutual agreement on a clear definition of the problem. Always check whether they want your help. Even when people agree that they want your help, that agreement is not usually a lifetime contract. People who want to help others generally expect to get something for themselves, though they may not be aware of it. Most people understand that helpers are selfish, but also think they are exceptions to the rule. Attempts to help are often

> interpreted as attempts to interfere. No matter how strange it may look, most people are actually trying to be helpful." (p. 135 – 137, Weinberg)

Leaders can face obstacles to effective organizing. "Organizing is not about solving problems, but avoiding them. The biggest obstacle to effective organizing is our eagerness to reward ineffective organizing. A problem-solving leader's entire orientation is toward creating an environment in which everyone can be solving problems, making decision, and implementing those decisions, rather than personally solving problems, making decisions, and implementing those decisions." (p. 203 – 204, Weinberg)

Chapter 5: Critical Thinking

The Socratic Method

The Socratic method is at the root of problem solving and questioning. This involves critical thinking and making inquiries. It is a structured logical thought process of asking questions. “What is the purpose of the Socratic method? To put the question in a more Socratic way: we wear glasses because we don’t see the external world clearly without them; we take x-rays to see inside the physical self. What is the Socratic method for? It lets us see something else more clearly: the workings and failings of the mind and its productions.” (p. 19, Farnsworth) The Socratic method gets parties to agree to what’s true while discarding what is untrue. “Socratic questioning is better viewed mostly as a way to think about hard questions on your own. Socrates starts with whatever you say—call it X. Then he gets you to admit that you also believe Y. Then he causes you to see for yourself that X and Y are inconsistent. Neither has been proven wrong, but at least one of them must be. Since you can’t believe both, you’re forced to change one or the other.” (p. 28, Farnsworth) Socrates taught people that philosophy was about examining oneself and others. “Thinking is, precisely, the inward dialogue carried on by the mind with itself without spoken sound. Writing out your own dialogues is a good way to sort out your own thinking and to develop ability with the Socratic method.” (p. 36 – 38, Farnsworth) The function of the Socratic method is to be a “truth-teller, a questioner of convention, an irritant. The Socratic method is disruptive. It exposes the truth and creates discomfort.” (p. 40, Farnsworth) The Socratic method is all about process rather than result. It is about asking questions to get to the truth.

Embracing the Socratic method encourages people to continue persevering in searching for answers. For programmers, or software engineers, the search for answers is part of the job, whether that search involves attempting different coding solutions or searching Google for answers. The Socratic method is an activity rather than a set of beliefs. It encourages people to test opinions. For software engineers, it can encourage the testing of code to ensure that the code works. Asking questions is the key rather than assuming that you have the answer and that the code works. When thought takes the form of questions, it makes the mind focus on finding answers to those questions rather than accepting frustrating answers. “Every time you ask and answer good questions, your understanding gets a bit deeper. You see more complexity.” (p. 45, Farnsworth) Asking questions forces a software engineer to develop more robust solutions rather than using the first quick answer that comes to mind. The constant questioning of an answer subjects it to testing that is necessary in the engineering process. This improves the quality of the software solution. The software engineer can proceed in steps to a solution. The software engineer wants to poke holes in a potential solution to ensure that it will solve the problem rather than creating new issues. Questioning “breaks reasoning down into clear steps. If something goes wrong, this makes it easier to figure out where.” (p. 47, Farnsworth) Complex solutions should not be hurried. The Socratic method can help a software engineer to understand a problem more completely rather than simply implementing the first idea that comes to mind. It’s about making code bulletproof. Software engineers must completely understand a problem prior to developing a robust solution. The Socratic method helps people ask the right questions, which is essential in software engineering when researching a problem and developing a solution.

Critical Thinking

“Critical thinking is a way of thinking about big and small questions you will face for as long as you’re alive. Critical thinking is a way of thinking about events and ideas instead of simply accepting them at face value. It is disciplined thinking that is clear, rational, open-minded, and informed by evidence. The ability to think clearly and logically is at the heart of critical thinking.” (p. 11 – 12, Thinknetic, The Socratic Way of Questioning) The attributes of the critical thinker include the following:

open-minded, analytic, systematic, inquisitive, judicious, truth-seeking, confident in reasoning. Critical thinking skills include the following: observation, reflection, analysis, evaluation, interpretation, inference, problem solving, decision-making, explanation. Critical thinking is "the ability to think about connected ideas thoroughly and independently, basing those ideas on factual evidence." (p. 13, Thinknetic, Critical Thinking & Logic Mastery) "Critical thinking is a highly valued intellectual ability that involves objectively analyzing and evaluating information before forming a judgment. It refers to the ability to look at issues objectively and make reasoned judgments free from emotional bias or unfounded assumptions." (p. 12, Schuster) Empathy and compassion can be a part of critical thinking. Critical thinking involves the following concepts (p. 18 – 21, Thinknetic, Critical Thinking & Logic Mastery):

- Perception: perceive the situation
- Assumptions: "unexamined beliefs taken for granted"
- Emotions: emotional indicators help critical thinking
- Language: "critical thinking is only possible when we make an abstract representation of reality through words"
- Arguments: "a well-reasoned list of assumptions and premises"
- Fallacies: "a belief or conclusion reached through unsound logic"
- Logic: "structured thinking, designed to evaluate information accurately"
- Problem solving: applying "time-consuming skills when confronted with a severe problem in need of resolution"

Logic is the basis for having a critical mindset. The elements of logic include propositions, premises, and conclusions (p. 23, Thinknetic, The Socratic Way of Questioning):

- “Propositions: a statement used as the foundation of a logical argument. The proposition is either true or false.”
- “Premises: also called claims, are used to build the argument and provide support for it.”
- “Conclusions: the result of the argument is drawn from the premises or is inferred in the premises.”

Deductive reasoning is a process of logical thinking, which uses a general-to-specifics approach to reasoning. Another process of logical thinking is inductive reasoning, which is a specifics-to-general approach to reasoning. To think critically, a person must be curious and be eager to listen to others. Critical thinkers need to question everything they read instead of accepting it with blind faith. It’s important for the critical thinker to have healthy skepticism. They should maintain objectivity when assessing a situation and develop a habit of questioning things. In some cases, characteristics of a group can be used to generalize about the larger population, but one must be careful not to over-generalize. “The antidote to over-generalization is thorough and deliberate critical thinking, promoting objectivity and resilience to fallacies.” (p. 33, Schuster) Barriers to critical thinking include the following:

- Egocentrism
- Sociocentrisms
- Inconsistencies
- Stubbornness
- Prejudice
- Fear
- Laziness

An individual can improve their critical thinking by doing the following:

- Take a break and think about the problem

- Manage emotions that come about during discussion of the problem
- Gather others' points of view
- Question the assumptions of what you do on a daily basis
- Gather facts and data to justify why you are doing what you are doing
- Interact with people outside your team or department

Avoid treating every issue like a crisis and prioritize tasks effectively. Act urgently when the situation calls for it but be more reflective during less urgent situations. It helps to think like a scientist to exercise critical thinking skills. Develop a healthy form of skepticism, or a questioning attitude. Investigate anomalies to help question assumptions. Treat assumptions as hypotheses and try to measure them rather than accepting them at face value. When thinking critically about a problem, it's important to use hard evidence to answer questions and develop solutions. Don't just accept cause and effect assumptions. Challenge them by testing them, such as changing variables and performing experiments. In essence, people in businesses should employ the scientific method when thinking critically about problems. To be more effective at critical thinking, change the way you see the world. Look at things more in-depth. "When we look at the world, we should not just examine, but examine with a deliberately different purpose." (p. 43, HBR Guide to Critical Thinking) When using critical thinking to solve problems, focus on problem diagnosis to ensure you are solving the right problems.

Critical thinking will involve data, and it is important to use it effectively when solving a problem. To use data effectively, it's necessary to ensure the quality of the data. Audit data transactions to determine the source of it and how it's been maintained. For critical thinking purposes, it is essential to develop purposeful data models.

Critical thinking means being able to change your mindset. It also means that creativity is essential to solving problems. This means being open-minded when solving problems. Critical thinkers must maintain intellectual humility, which allows them to change their minds when confronted with data or ideas that contradict their assumptions and currently held ideas. Additionally, intellectual humility means that the critical thinker may not need to change his or her idea if evidence confirms its validity. Intellectual humility can be broken down into the following components (p. 105, HBR Guide to Critical Thinking):

- "Having respect for other viewpoints"
- "Not being intellectually overconfident"
- "Separating one's ego from one's intellect"
- "Being willing to revise one's own viewpoint"

To have intellectual humility, it is essential to be a good listener.

"Critical thinking requires not only an awareness of a current situation but also an understanding of what has worked in the past, what hasn't, and how that can affect future decisions. But doing so requires a regular practice of assessment and reflection." (p. 195, HBR Guide to Critical Thinking) Critical thinking involves thinking about failure, learning from it, and using it in future analyses. Critical thinking involves the need to learn. Learning about a problem, situation, circumstances, people, and more contributing factors is essential to the critical thinking process.

"Dr. Richard Paul and Dr. Linda Elder introduced the concept of Universal Intellectual Standards to guide critical thinking." (p. 54, Schuster) The Universal Intellectual Standards include the following (p. 55 – 56):

- Clarity: "asking for more details or examples to understand the information better"
- Accuracy: "verifying the information and cross-checking with other sources"
- Precision: "asking for more specific or detailed information"

- Relevance: “how the information connects to the problem and helps solve it”
- Depth: “the challenges and difficulties associated with the problem”
- Breadth: “considering other points of view or perspectives”
- Logic: “assessing if the information aligns with other information and fits into the bigger picture”
- Significance: “determining the most important aspects to consider”
- Fairness: “ensuring all perspectives have been heard and considered equally, without altering facts to favor certain opinions”

Paul and Elder identified the elements of thinking, which improve critical thinking (p. 59 – 60):

- Purpose: “thinking is goal-oriented”
- Question: “the question driving the thinking process should be precise and clear”
- Assumptions: “reassess assumptions to ensure they are evidence-based periodically”
- Point of View: “all points of view must be considered equally during analysis”
- Information: “the data gathered to solve a problem or answer a question must be accurate, relevant, and evidence-based”
- Inferences: “conclusions drawn must be based on evidence and facts and should be logical and consistent”
- Concepts: “theories and ideas generated must be justified and alternative explanations must be explored”
- Implications and Consequences: “consider all possible outcomes before deciding on a course of action”

Paul and Elder determined valuable intellectual traits that include the following (p. 63 – 72):

- Intellectual humility: “acknowledgement of our human limitations in terms of knowledge”

- Intellectual courage: "the strength to confront ideas and beliefs that may invoke strong negative emotions rather than shying away from them"
- Intellectual empathy: "immersing oneself in another person's perspective to accurately comprehend their thoughts and views"
- Intellectual autonomy: "owning one's thinking process and being accountable for it"
- Intellectual integrity: "the ideal of being authentic to one's thought processes"
- Intellectual perseverance: "epitomizes the unwavering commitment to critical thinking, irrespective of the trials and tribulations encountered"
- Confidence in reason: "an earnest conviction that, in the long run, truth and reason will prevail"
- Fair-mindedness: "a trait that demands impartiality equitable judgment, and respectful attention to different perspectives"

Bloom's Taxonomy (see Figure 4) presents critical thinking as a deliberate and structured process. It is composed of the following levels (p. 39 – 44, Thinknetic, Critical Thinking & Logic Mastery):

- Remember: remember relevant forms and sources
- Understand: "study the relevant materials until you have a full understanding of the issue"
- Apply: "look at the knowledge you gathered while working through the first two rungs of the pyramid"
- Analyze: "break the problem into parts
- Evaluate: "look at your analysis carefully and evaluate it"
- Synthesize: "take all the elements arrived at and unite them into one plan"
- Create: implement the plan

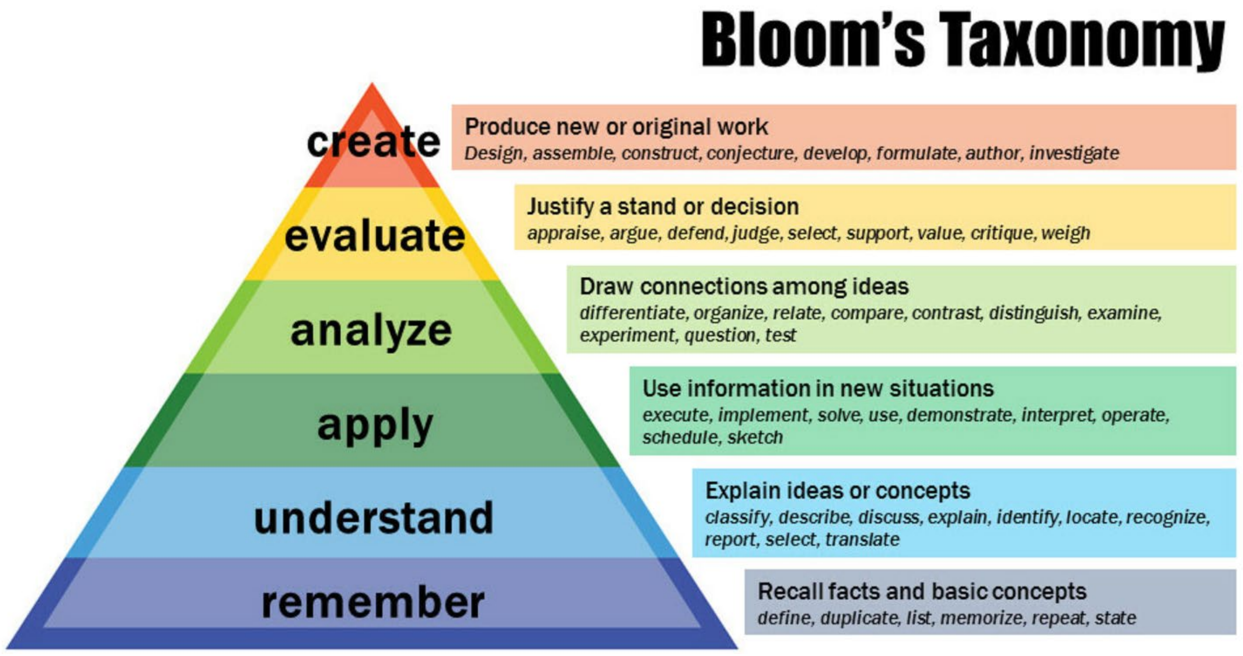
Bloom's Taxonomy
create
Produce new or original work
Design, assemble, construct, conjecture, develop, formulate, author, investigate
evaluate
Justify a stand or decision
appraise, argue, defend, judge, select, support, value, critique, weigh
analyze
Draw connections among ideas
differentiate, organize, relate, compare, contrast, distinguish, examine, experiment, question, test
apply
Use information in new situations
execute, implement, solve, use, demonstrate, interpret, operate, schedule, sketch
understand
Explain ideas or concepts
classify, describe, discuss, explain, identify, locate, recognize, report, select, translate
remember
Recall facts and basic concepts
define, duplicate, list, memorize, repeat, state

Figure 4 Bloom's Taxonomy (Source: Vanderbilt University)

Leaders demonstrate the ability to think critically and analyze. They figure out how to incorporate technology that appropriately benefits the organization. They lead teams that build and develop systems that automate processes. Critical thinking is essential for leaders to exercise when making decisions. For leaders, one aspect of critical thinking is to eliminate false assumptions. “False assumptions are the main cause of projects that fail. Managers, perfectly able to qualify an assumption as bad, build doomed plans on hidden false assumptions without even acknowledging their existence.” (p. 13 – 14, Atanasiu) When leaders or managers make assumptions, it is essential to discuss those assumptions with others and make a list of those assumptions. Leaders need to think in advance of reasons for possible failures to safeguard against them. It is critical for leaders to test their assumptions before implementing their plans. Assumptions can be tested by carrying out experiments.

Chapter 6: Engineering

In a business or technical setting, problem solving is often accomplished by the process of engineering a solution to the problem. “Engineering is the application of scientific principles toward practical ends.” (p. 33, McConnell, Professional Software Development) Practical applications in engineering advance faster than the science does that backs them up. “Engineers learn what is true, what is useful, and how to apply well-understood knowledge to solve practical problems.” (p. 30, McConnell, Professional Software Development) “Professional engineering requires in-depth knowledge of both theory and practice.” (p. 178, McConnell, Professional Software Development)

Engineers need to understand minute technical details. As engineers gain more experience and knowledge, they need to see the big picture of complex designs. More experienced engineers need to understand the architecture of systems. In addition, engineers need to understand the context of the situations or problems that they face. The context could be what the organization needs or any number of factors. Engineers, as they gain experience on projects, need to be aware of budgetary constraints. The needs of the organization influence an engineer’s choices. Engineers provide technical insight that helps managers and leaders make decisions. Oftentimes, experienced engineers act as technical architects and lend their professional expertise to the overall decision making of the organization. Experienced engineers become leaders on project teams. As engineers proceed in their careers, they will need to gain both technical skills and nontechnical skills. Nontechnical skills include working with others, communication, mentoring, and more.

Engineers are responsible for designing systems that people use. It’s essential that those systems are designed with human factors in mind. Safety, security, and reliability are just some of the criteria that engineers must consider when designing systems. Engineering failures can be disastrous for people, whether they are software bugs, plane crashes, or malfunctioning equipment. Engineers must be concerned with the smallest details as well as the overall blueprint of designs. Engineers must take the quality of the systems they develop seriously. Engineers need to be strategic about their technical designs. Each detail needs to be thought out thoroughly.

Engineers need to be mindful of time constraints. It's healthy to maintain work-life balance. Like everyone else, engineers have personal commitments. Engineers should adhere to the concept of opportunity cost. "Everything you commit to has an opportunity cost. By choosing to do one thing, you're implicitly choosing not to do another." (p. 116, Reilly)

Engineers should care about the customer's experience because the systems they build are ultimately being used by those customers. Engineers should care about the usability of their systems. Engineers need to know how the success of a system will be measured. These metrics should be included in the requirements documentation.

Engineers build models to understand complexity. Sometimes those models are prototypes that are also used to demonstrate potential system features to stakeholders. To build models, engineers must abstract the details in an organized manner. For engineers, it's important to use the models they build to clarify the details that are relevant to developing the system. It's important for engineers to document in some way the concepts that can be derived from the models to be used for building systems.

Engineering requires strong leadership that is not always taught at colleges and universities. Engineers get promoted to leadership positions, without knowledge of the various qualities that leaders need to successfully guide teams. Communication skills and ethics are important components of the engineering profession. Leadership and teamwork are also essential skills for engineers. As engineers get promoted within organizations, they spend an increasing amount of their time on non-technical tasks, such as managing subordinates and communicating with other members of the organization.

Before engineers can effectively lead others within their organizations, they must be able to manage or control themselves. Managing people requires a different skillset than the technical education that engineers received at colleges and universities. “The top five attributes that employers look for in job candidates are leadership, the ability to work in a team, written communication skills, problem-solving skills, and verbal communication skills.” (p. 8, Hess) An important factor between leaders and members of an organization is the influence that the leaders have on their subordinates, their peers, and their bosses. “When a leader has assembled the correct team members, and challenges them according to their capabilities and strengths, micromanaging is seldom necessary; employees will be self-motivated. Effective leaders communicate concerns and difficulties that might be encountered when undertaking complex or challenging tasks or project.” (p. 23, Hess) Engineers, who want to lead effectively, use the following behaviors and strategies (p. 25, Hess):

- “Display confidence, courage, optimism, passion, responsibility, and accountability”
- “Are available and approachable”
- “Show kindness, consideration, respect, and appreciation for others”
- “Take advantage of insight/knowledge gained from successes and failures”
- “Clearly define and communicate objectives and goals”
- “Ask for assistance and additional resources when needed”
- “Offer advice, evaluate performance, develop employees professionally and personally and help employees work through problems”
- “Acknowledge effort and achievement routinely”
- “Criticize diplomatically without blame or anger”
- “Admit when they are wrong and correct the situation”
- “Are transparent, authentic, professional, unbiased, and ethical”

- “Keep an open mind for alternative/creative views of approaches and seek participation from team members in decisions, i.e. they listen”
- “Ask numerous questions to gain understanding and show involvement; they listen carefully to the responses to show their interest in team member opinions and ideas”

The skills possessed by successful technical leaders include the following:

- Behaves ethically and professionally
- Manages change
- Influences others
- Takes risks
- Communicates clearly and often
- Makes timely decisions
- Shows resilience
- Builds teams
- Technical competency
- Vision; significant problem identification
- Continuous Learning
- Passion for science and engineering

The technical leader must exhibit a growth mindset. A growth mindset is “where individuals seek and accept challenges, perseveres in the face of obstacles, and use criticism to improve; that is, they are driven to learn and are optimistic.” (p, 37, Hess)

Design Mistakes

Engineering requires learning from successful examples of engineering projects as well as failures that resulted from the engineering process. Software engineers will do well to learn about design mistakes in other engineering disciplines to take seriously the need to be detail oriented. Preventing failure in engineering designs is difficult. Christopher Alexander, who wrote many classic books on engineering/architecture wrote the following:

> "We are never capable of stating a design problem except in terms of the errors we have observed in past solutions to past problems. Even if we try to design something for an entirely new purpose that has never been conceived before, the best we can do in stating the problem is to anticipate how it might possibly go wrong by scanning mentally all the ways in which other things have gone wrong in the past." (p. 3, Petroski)

Engineers need to design with a paranoid sense of the potential mistakes that could happen to prevent them from happening. Lev Zetlin stated, "Imagination and fear are among the best engineering tools for preventing tragedy." (p. 3, Petroski) Design failures have been recognized throughout history in civil engineering and mechanical engineering projects, but they can happen in any engineering discipline. Software can be more complex than bridges, but software engineers can learn lessons from civil engineers. In an article in a journal of the Association for Computing Machinery (Spector and Gifford, 1986), the following was written: "Though some computer systems are more complex than even the largest bridges, there is a wealth of experience and insight in the older discipline that can be of use in computer systems designs, particularly in such areas as specification, standardization, and reliability." (p. 4 – 5, Petroski)

Some have argued that human error is the root of all engineering failures. "Human error in anticipating the future continues to be the single most important factor in keeping the reliability of engineering designs from achieving the theoretically high levels made possible by modern methods of analysis and materials. Education, motivation, and quality control provide clear, if not necessarily easily implemented, ways of reducing human error in manufacturing, construction, and maintenance, but ways to eliminate human error from the design process are much less obvious." (p. 7 – 8, Petroski) Quality control (also called quality assurance or quality engineering) is one method of reducing design failures. A team of individual quality control testers should be dedicated to detecting flaws in design. When designing something that is not based in common practice, it's important to proceed with caution.

While trying to satisfy a particular aspect of design, such as aesthetics or economics, functional design may be overlooked. It is incumbent upon the engineer to focus on functionality prior to working on other aspects of design. Engineers should be aware of Parkinson's Law of Triviality (also known as bikeshedding). It's the concept that people spend an inordinate amount of time devoted to unimportant details rather than focusing on the most critical parts of a project. The focus of a project and design should be on functionality and usability. Engineers should avoid hubris, or arrogance, when creating a design. Instead, engineers should exhibit a quiet confidence that is achieved through thousands of hours of working hard at creating designs and learning from mistakes.

Engineers need to take the time necessary to create functional systems. If a problem is encountered in a design, it is incumbent on the engineer to take a step back and correct the error. Building a model, or prototype, can help an engineer prevent errors in design. For a software engineer, understanding the scaling needs of a system is essential to handle capacity of processing data and user input or storage of data in databases. The details of software design sometimes require revisiting different aspects of what is laid out in requirements documentation. A change in design can cause a failure in another part of the system. Thus, regression testing software is essential to ensure that the entire system works as intended. The programming of software can introduce logic errors, which are different from errors detected by the compiler. Logic errors may not become evident until a later stage of development or even once the software goes into production. Quality assurance and use cases are necessary to prevent logic errors from appearing in production software. Unit tests are also effective at detecting logic errors. Software engineers create designs based on judgment gained from many hours of developing software according to specifications. It's judgment that prevents mistakes.

Chapter 7: Efficiency and Quality

Efficiency

Individuals and organizations are constantly trying to be more efficient. Efficiency is about accomplishing more than what was accomplished before with fewer resources.

In manufacturing organizations, the just-in-time (JIT) manufacturing system involves receiving supplies just in time to use them and providing inventory immediately before it is needed. JIT systems promote efficiency and force organizations to be lean. Demand drives the efforts of the JIT system.

Quality and Quality Control

Quality control (QC) must be utilized to reject input that is defective before it is allowed to affect the product or service. Put quality control first.

Continual Improvement Process

In engineering, the goal of continual improvement process is aimed at eliminating waste and defects, while preventing errors and improving efficiency and productivity. It's necessary to define productivity first. "Productivity is meaningless unless you know what your goal is." (p. 32, Goldratt and Cox) One must determine what they are supposed to be doing. Then, identify the goals that have already been established to help identify the actual goal. Next, identify actions that are productive and actions that are nonproductive. Understand the productivity achieved compared to the resources that were invested. Once the goal is understood, every action taken should be a means to achieve the goal. The goal remains the same but can be restarted with measurements to determine if the goal is being achieved. Recognize bottlenecks that are preventing reaching the goal. Bottlenecks “are any resource whose capacity is equal to or less than the demand placed upon it.” (p. 139, Goldratt and Cox) Also, recognize non-bottlenecks, which are “any resources whose capacity is greater than the demand placed on it. Make the flow through the bottleneck equal to demand from the market.” (p. 139, Goldratt and Cox) In order to manage bottlenecks, prioritize the order of tasks to complete. A system is needed to prioritize tasks. Communicate about the priorities and the need for a system.

Kaizen

Six Sigma

Lean Manufacturing

Chapter 8: Toyota

The Toyota Production System is lean manufacturing; it focuses on process, people, and technology. Rival automakers adopted lean manufacturing pioneered by Toyota. The Toyota Lean Production Development System is shown in Figure 5. Lean manufacturing can be extended to departments of entire organizations. Toyota vehicles are known for their quality and rank high in J.D. Powers surveys. "The only way a company can make significant breakthrough improvements in product development performance is to build its own product development system with the patience and philosophical underpinnings that has led to the success of Toyota and other great companies." (p. 12, Morgan and Liker)

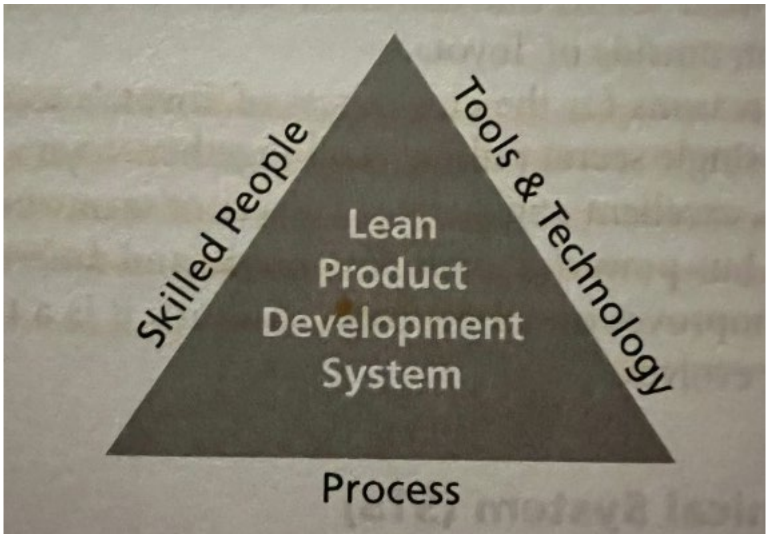

Figure 5 Toyota Lean Production Development System (Source: The Toyota Product Development System)

The Lean Product Development System Model is composed of the following (p. 15 – 24, Morgan and Liker):

- Process
  - Establish customer-defined value to separate value-added activity from waste
  - Front-load the product development process while there is maximum design space to explore alternative solutions thoroughly

  - Create a leveled product development process flow
  - Utilize rigorous standardization to reduce variation, and create flexibility and predictable outcomes
- People
  - Develop a chief engineer system to integrate development from start to finish
  - Organize to balance functional expertise and cross-functional integration
  - Develop towering technical competence in all engineers
  - Fully integrate suppliers into the product development system
  - Build in learning and continuous improvement
  - Build a culture to support excellence and relentless improvement
- Tools and technology
  - Adapt technology to fit your people and processes
  - Align your organization through simple, visual communication
  - Use powerful tools for standardization and organizational learning

Toyota focuses on the process of product development first. “No effort or resources should be expended without a deep understanding of customer-defined value.“ (p. 28, Morgan and Liker) At Toyota, the Chief Engineer establishes the concept for engineers to follow in designing vehicles and customer value is the focus. Like software, automobiles become expensive if changes are made to the design later in the product development process. Following the Agile project management process can allow changes to be made in design as the software is delivered in incremental steps. Toyota shares resources across concurrent projects to manage technological complexities. The company uses common standard vehicle platforms to design their automobiles, which contributes to vehicle quality and reliability. In software engineering, design patterns exist to provide programmers with standard templates for building software. “Toyota engineers have a strong sense of the vehicle as a system and consequently focus a great deal of skill and energy at the design interface." (p. 50, Morgan and Liker) Toyota engineers study several alternative designs before building new automobile models.

Toyota uses value stream mapping to eliminate waste in the product development process. Lean manufacturing looks "at the total value stream mapping because waste between process steps is likely to be far greater than waste within a single process step." (p. 70, Morgan and Liker) Toyota avoids overburdening engineers, or muri, because overburdening results in work quality suffering and safety risks happening. Toyota works to control the variability in the product development process. "Designing and developing an automobile takes hundreds of engineers, thousands of components and tools, and high-tech equipment to manufacture each of those components." (p. 90, Morgan and Liker)

Toyota utilizes rigorous standardization to reduce variability and create flexibility and predictable outcomes. The company uses checklists, which are very helpful at keeping track of the engineering process. "A best solution cannot necessarily be predicted in advance. It is learned over time through experience and is guided by the spirit of kaizen, which postulates that there is always an opportunity to learn more and that learning is an ongoing process. The spirit of engineering kaizen is driven by the never-ending pursuit of technical excellence that underlies consistent checklists utilization, validation, and improvement." (p. 102, Morgan and Liker) Toyota standardizes the process of product development to reduce variability. One of the benefits "of a standard development process is that it contributes to more precise communication and greater understanding across engineering organizations by providing a common framework for discussion." (p. 106, Morgan and Liker) Toyota's engineering organizations work together collaboratively "to create effective process and design standards and a culture of discipline." (p. 111) Toyota puts its engineers through an eight-year developmental period when they join the company.

At Toyota, the Chief Engineer oversees design projects and serves as project manager. The Chief Engineer and their staff bear the following responsibilities:

- Voice of the customer
- Customer-defined value
- Product concept
- Program objectives
- Vehicle-level architecture
- Vehicle-level performance
- Vehicle-level characteristics
- Vehicle-level objectives
- Vision for all functional program teams
- Value targets
- Product planning
- Performance targets
- Project timing

Toyota used creativity, challenge, and courage in the development process of the Lexus and the Prius. When researching the market for the Lexus, the Chief Engineer reviewed information from a couple of focus groups rather than from marketing surveys. Lexus Chief Engineer Ichiro Suzuki asked the following questions when thinking about the viability of Lexus (p. 123, Morgan and Liker):

- “What does it mean to own a high-quality luxury vehicle?”
- “What characteristics does a car need to make people who own it feel like they are wealthy—financially and emotionally?”

- “What characteristics would a car have that, as the years go on, would make people feel more attached to it and possessive of it?”

The Lexus was developed with exceptional performance and elegant appearance in mind. For the Prius, Toyota executives chose a Chief Engineer, Takeshi Uchiyamada, who “had to invent a new method for developing cars.” (p. 126, Morgan and Liker) The development team decided to take natural resources and the environment into consideration for the Prius. Uchiyamada got the development team to be unconventional when making the interior of the Prius more spacious.

Toyota is organized to balance functional expertise and cross-functional integration. “The secret to Toyota’s success is combining a strong functional organization based on deep specialization with the Chief Engineer system as the other leg of the matrix.” (p. 141 – 142, Morgan and Liker) The matrix organization provides the following (p. 142, Morgan and Liker):

- “An excellent balance of functional expertise and cross-functional integration”
- “The technical depth and efficiency of the functional organization with the customer focus of the product organization”
- “Flexibility in assigning resources to programs as well as technical depth in responding with creative solutions to new problems”

Toyota maintains an intense customer focus. The company is able to integrate the functions and programs because of the following:

- The customer is first
- The Chief Engineer is revered
- The Chief Engineer has the power of executive sponsorship

- General managers understand the importance of serving customers and cross-functional cooperation
- Junior employees respect senior employees

As complexity in building vehicles increased, Toyota reorganized around different vehicle platforms. To handle complexity, Toyota created the modular development team (MDT) and the chief production engineer role.

Toyota develops towering technical competence in all its engineers. Toyota engineers must have integrity and competence. "Toyota's engineers can do everything from designing tooling to developing complete production equipment and overseeing its construction." (p. 165, Morgan and Liker) "People development at Toyota seems to be just as important as product development. Toyota managers are trained to be teachers and see every engineering project as an opportunity for developing its engineers." (p. 169, Morgan and Liker) When a new engineer joins Toyota, they are paired with a mentor (a senior engineer) and assigned an improvement project. Toyota emphasizes genchi genbutsu, which means keeping engineers close to the product to understand deeply the engineering situation. Like Daily Standup Meetings used in software development, Toyota has daily build wrap-up meetings to gather the "Chief Engineer, body engineers, prototype technicians, production team leaders, production engineers, and suppliers at the end of each day" to discuss issues encountered that day. (p. 175, Morgan and Liker)

Toyota fully integrates suppliers into the product development system. The company "makes sure that every car part reflects Toyota quality by making every supplier an extension of Toyota's product development process and lean logistics supply chain." (p. 180, Morgan and Liker) Toyota works with suppliers "on cost reduction, even offering to help suppliers change the product design through value engineering." (p. 182, Morgan and Liker) "Toyota has established a great level of professional trust with its suppliers, closely paralleling the trust that Toyota engineers have established with each other." (p. 182, Morgan and Liker) The suppliers for Toyota end up becoming an interlocking set of corporations. Toyota teaches suppliers to improve their products rather than rejecting them. Toyota slowly integrates suppliers into the product development process. Toyota outsources over 70% of its vehicle content.

Toyota builds in learning and continuous improvement. "Toyota consistently worked on developing ways to collect, disseminate, and apply tacit knowledge, creating a learning network that is applied enterprise-wide from product development to manufacturing." (p. 205, Morgan and Liker) Toyota forms its learning network by doing the following:

- Supplier technology demonstrations
- Competitor teardown analysis
- Checklists and quality matrices
- Learning focused problem solving
- Know-how database
- Hansei (reflection) events
- Program manager conferences
- Business revolution teams
- On-the-job training skills matrices and learning-focused career paths
- Resident engineers within Toyota and with affiliated companies

Hansei, or reflection, at Toyota takes place via the following methods:

- Personal reflection
- Real-time reflection
- Postmortem reflection

Toyota sees technical problems as part of the product development experience. Cross-checking is a method used by Toyota "to discover problems and check quality." (p. 212, Morgan and Liker) Toyota uses daily wrap-up meetings to capture "lessons learned, clarify assignments, and generally aid in real-time, course-correction decisions." (p. 213, Morgan and Liker)

Toyota has built a culture to support excellence and relentless improvement. A definition of culture is:

- "The pattern of basic assumptions that a given group has invented, discovered, or developed in learning to cope with its problems of external adaptation and internal integration, and that have worked well enough to be considered valid, and, therefore, to be taught to new members as the correct way to perceive, think and feel in relation to those problems." (p. 218, Morgan and Liker)
- Culture operates at an unconscious level.
- Culture involves necessity-driven and empirically-based production systems.
- Culture involves adapting systems as necessary.
- Culture involves learning the system by doing.
- "A strong culture is one in which many things are shaped by a large cross-section of people." (p. 220, Morgan and Liker)

"Toyota works hard to bridge differences, and as a result, most Toyota employees do share basic assumptions about values, priorities, and how to get work done." (p. 220, Morgan and Liker) Tools must be adapted to the culture of an organization. "Toyota has a long history of training engineers in rigorous methods to collect information, identify real problems, get at critical essentials, consider multiple potential solutions, get broad input through nemawashi, creatively develop root cause countermeasures, and use discipline in implementing. Once all of this is in place, Toyota engineers follow up with standardization and continuous improvement." (p. 221, Morgan and Liker)

Toyota's culture is focused on customers first. "Toyota's cultural structure includes the following precepts:" (p. 222, Morgan and Liker)

- "Run by engineers, making it a technical hierarchy"
- "Fundamentally, a manufacturing company that translates into core value-added manufacturing activities and those supporting manufacturing"
- "Focuses on developing technical mastery using the 'hands-on' Toyota scientific method"
- "The individual engineer is central to the product development system"
- "Learning and continuous improvement (every-day kaizen) is fundamental to how work gets done"
- "Process discipline, hard work and loyalty is expected of everyone (especially leaders)"
- "Data focused"

Fujio Cho became president of Toyota in 1999 only after many years of training in the Toyota Production System (TPS). Toyota engineers are expected to work hard, work long hours, demonstrate significant technical knowledge, and be passionate about their work." (p. 224, Morgan and Liker) "Toyota values discipline and work ethic and requires these of everyone—inside and outside the company." (p. 225, Morgan and Liker) The discipline expected at Toyota is seen in the following:

- Standardization and working to process
- Maintaining schedules
- A3 disciplined communication method (a visual one-page report)
- Nemawashi (the process of getting consensus while the task is in progress)

Toyota embraces kaizen, which is commitment to improvement every day. It is essential to be humble when embracing kaizen. Toyota emphasizes customer first in its culture. Toyota believes that having a culture of learning is essential to having a competitive advantage. Toyota embraces genchi genbutsu, which involves reviewing defective parts to improve quality. At Toyota, the focus is on the team and credit goes to the team rather than the individual. Everyone at Toyota is accountable and has the willingness to accept responsibility for failure (this is the spirit of hansei). To align employees with their teams and the organization, Toyota employs the hoshin planning process, which "determines goals company-wide; every employee has objectives that are developed with his or her immediate supervisor, and these objectives dovetail with the objectives of the next level up." (p. 231 – 232, Morgan and Liker) "Toyota is about the process. Follow the right process and you will get the desired result." (p. 235, Morgan and Liker).

Another part of Toyota's product development system is tools and technology. Adapt technology to fit your people and processes:

- Technologies must be seamlessly integrated
- Technologies should support the process, not drive it
- Technologies should enhance people, not replace them
- Specific solution oriented: not a silver bullet
- Right size—not king sized
- Adapting technology to enable process
    - Toyota focuses first on process before technology

Align your organization through simple, visual communication. The Chief Engineer produces a concept paper that "defines the core parameters of the entire product development proposal at Toyota." (p. 260, Morgan and Liker) "The concept paper is highly confidential, typically runs 15 to 25 pages in length, and text is supplemented by tables, graphs, and sketches intended to provide the team with a single unifying direction and decision-making guideline." (p. 260, Morgan and Liker) The lean view of communication involves the following (p. 261, Morgan and Liker):

- "If everyone is responsible, no one is responsible."
- "If everyone must understand everything, no one will understand anything very deeply."
- "If all communication is going to everyone, no one will focus on the most critical communication for their role and responsibility."
- "If you inundate your people with reams of data, no one will read it."

Toyota employs obeya, or "big room", to gather engineers in one place to design vehicles. Toyota uses the ringi system, which is a formal decision-making process used for handling significant decisions." (p. 265, Morgan and Liker) Toyota utilizes hoshin management, which "is an effective tool for aligning an organization toward the achievement of broader goals or objectives and allowing that organization to react quickly to a changing environment." (p. 266, Morgan and Liker) The phases of hoshin are the following:

- Strategic planning
- Hoshin deployment
- Controlling through metrics
- Check and act

Toyota uses the A3 problem-solving tool, which "refers to a standardized communication format, a disciplined process of expressing complex thoughts accurately on a single sheet of paper." (p. 269, Morgan and Liker)

Toyota uses powerful tools for standardization and organizational learning. Toyota maintains a know-how database. The know-how database is composed of standards checklists used by Toyota for over 40 years. In the know-how database, "engineering guidelines, graphics, and practices are outlined and explained in greater detail than was possible on handwritten checklists. Information is arranged by vehicle/part of just part type, and within the part-type database, there is a competitor benchmark component that allows engineers to view pictures of competitor products and teardown analysis." (p. 282, Morgan and Liker) Toyota uses set-based concurrent engineering (SBCE), which is "a powerful source of knowledge and continuous improvement, a cornerstone of Toyota's product development process." (p. 283, Morgan and Liker) Toyota uses trade-off curves, which is "a relatively simple tool that is consistently used by Toyota engineers to understand the relationship of various design characteristics to each other." (p. 284, Morgan and Liker). Toyota uses decision matrices, which are used to consider design alternatives. Toyota uses competitor teardown and analysis sheets where module development teams (MDTs) analyze specific parts of competitor vehicles and prepare analysis sheets for assistance in building Toyota vehicles.

Chapter 9: Software Engineering

"The most fundamental problem in computer science is problem decomposition: how to take a complex problem and divide it up into pieces that can be solved independently." (p. vii, Ousterhout)

Software engineering should utilize an engineering process just as much as any other engineering discipline uses. It is more than simply writing code and deploying that code to a production environment. Quality assurance is essential to ensuring that the code does not cause any harm or breakdowns of the software applications. Software engineers need to be detail-oriented just as much as engineers in other disciplines need to be.

Software Development Lifecycle (SDLC)

Amazon Web Services (AWS) defines the Software Development Lifecycle with the following:

> "The software development lifecycle (SDLC) is the cost-effective and time-efficient process that development teams use to design and build high-quality software. The goal of SDLC is to minimize project risks through forward planning so that software meets customer expectations during production and beyond. This methodology outlines a series of steps that divide the software development process into tasks you can assign, complete, and measure."

The SDLC is composed of the following phases:

- Planning
    - Planning means project planning and allocation of resources for the development of the software.
- Analysis and Design

    - Requirements are determined and the design of the software is conducted.
- Implementation
    - Implementation means programming and unit testing.
- Testing
    - Testing is for quality assurance, or system-wide testing, user acceptance testing, and regression testing.
- Deployment
    - Deployment means releasing changes to an environment after testing is completed.
- Maintenance
    - Maintenance involves resolving issues that are found after release of the software.

One of the goals of software development is reuse. Parts of projects that can be reused include the following:

- Software architecture
- Design patterns
- Reusable components
- Requirements
- User interface elements
- Project plans
- Test cases
- Technical review procedures
- Source code
- Configuration management controls

- Team structures

Software Requirements

Requirements can be defined by the following. Ian Sommerville and Pete Sawyer stated, “Requirements are a specification of what should be implemented. They are descriptions of how the system should behave, or of a system property or attribute. They may be a constraint on the 88development process of the system.” (p. 6, Wiegers and Beatty) Types of requirements include the following:

- Business requirements
- Business rules
- Constraints
- External interface requirements
- Features
- Functional requirements
- Nonfunctional requirements
- Quality attributes
- System requirements
- User requirements

Determining the requirements of software is key to delivering a successful software solution. The drawbacks of not determining the correct requirements up front include bugs being created in development, which might even surface in production. If the end customer can be involved in requirements gathering, software engineers are more likely to develop the features of the system that are expected. Business analysts typically draft requirements documentation and are primarily responsible for gathering requirements. The business analyst's tasks include the following:

- Define business requirements
- Plan the requirements approach
- Identify project stakeholders and user classes
- Elicit requirements
- Analyze requirements
- Document and communicate requirements
- Lead requirements validation
- Facilitate requirements prioritization
- Manage requirements

Successful requirements gathering involves getting input from end users, software engineers, and quality assurance professionals. The levels of requirements are:

- Business
- User
- Functional

Agile projects maintain requirements in user stories in a product backlog. Each iteration of a project is planned by selecting user stories for development during the iteration. The user stories are selected based on prioritization. Changes that are requested during an iteration are added to the product backlog for later development. Iterative development, such as Agile, allows changes to be incorporated within a project more easily than the traditional Waterfall model. In Agile projects, a product owner represents the needs of the users and assumes responsibility for prioritizing the product backlog. The product owner "straddles the project champion and business analyst functions, representing the customer, defining product features, prioritizing them, and so forth." (p. 115, Wiegers and Beatty)

"Business requirements refers to a set of information that, in the aggregate, describes a need that leads to one or more projects to deliver a solution and the desired ultimate business outcomes. Business opportunities, business objectives, success metrics, and a vision statement make up the business requirements." (p. 78, Wiegers and Beatty) It's important to prioritize requirements to deliver the most important software features first in projects. Product backlogs help to prioritize requirements. Requirements documentation should provide the following:

- Business opportunity
- Business objectives
- Success metrics
- Vision statement
- Business risks
- Business assumptions and dependencies
- Scope and limitations
- Major features
- Project priorities
- Software quality metrics

Requirements documentation should spell out the different roles that users play in the use of the software. Software requirements are gathered through the following methods:

- Interviews
- Workshops
- Focus groups

- Observations
- Questionnaires
- System interface analysis
- User interface analysis
- Document analysis

Use cases and user stories are very useful for documenting user interaction with the system. "A use case is a written description of how a user will perform tasks on your website. It outlines, from a user's point of view, a system's behavior as it responds to a request. Each use case is represented as a sequence of simple steps, beginning with a user's goal and ending when that goal is fulfilled." (Usability.gov) A simple use case would be something like the following:

Use Case Title	Add items to a shopping cart
Actor	Customer
Basic Flow	1. User browses merchandise 2. User selects product 3. User views product 4. User adds product to shopping cart

Use cases can also include diagrams, such as the following:

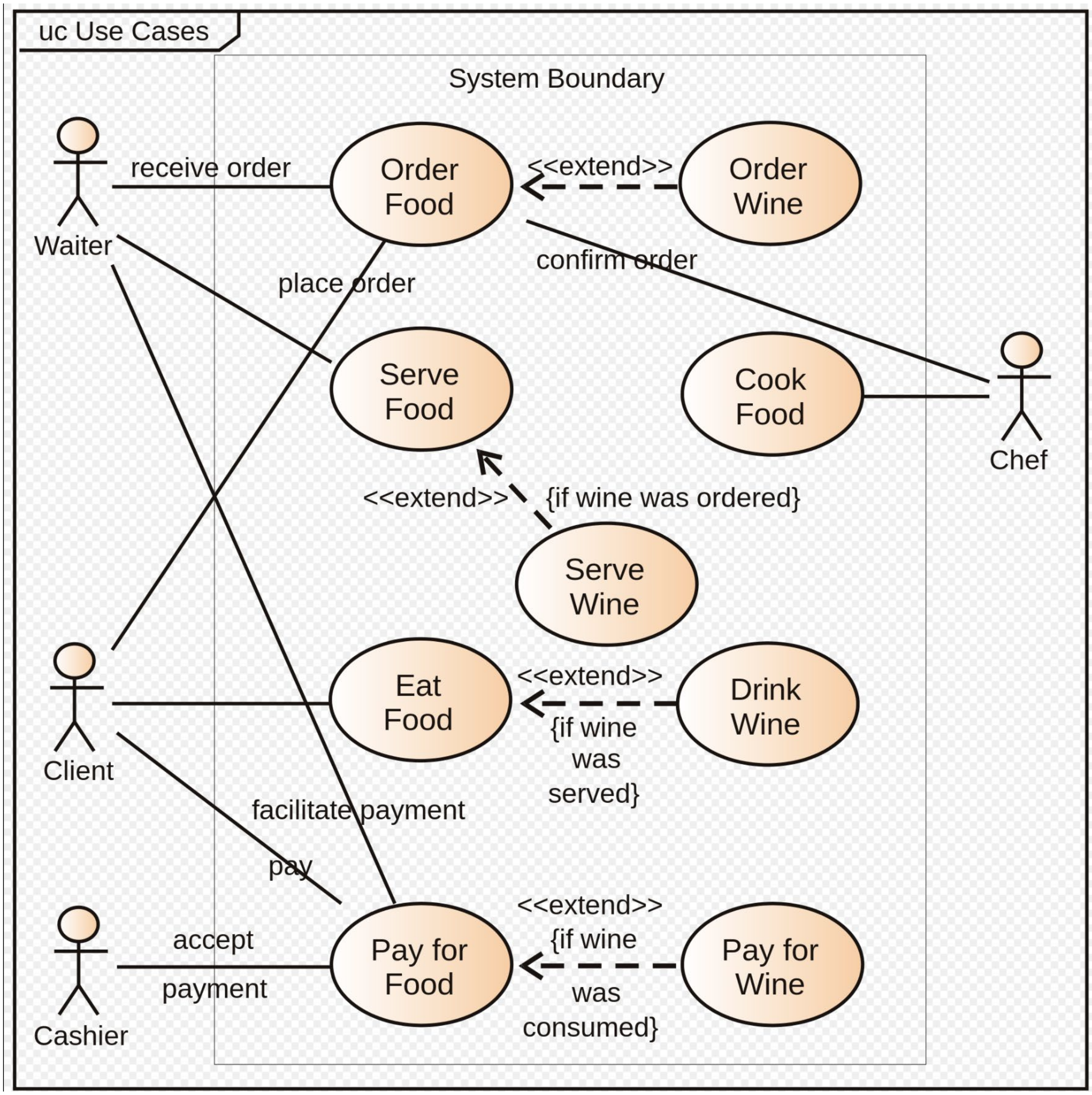

Figure 6 Use Case Diagram (Source: Wikipedia)

Software requirements can also spell out requirements of data flow for software. Data flow diagrams are useful for understanding how information is processed in software.

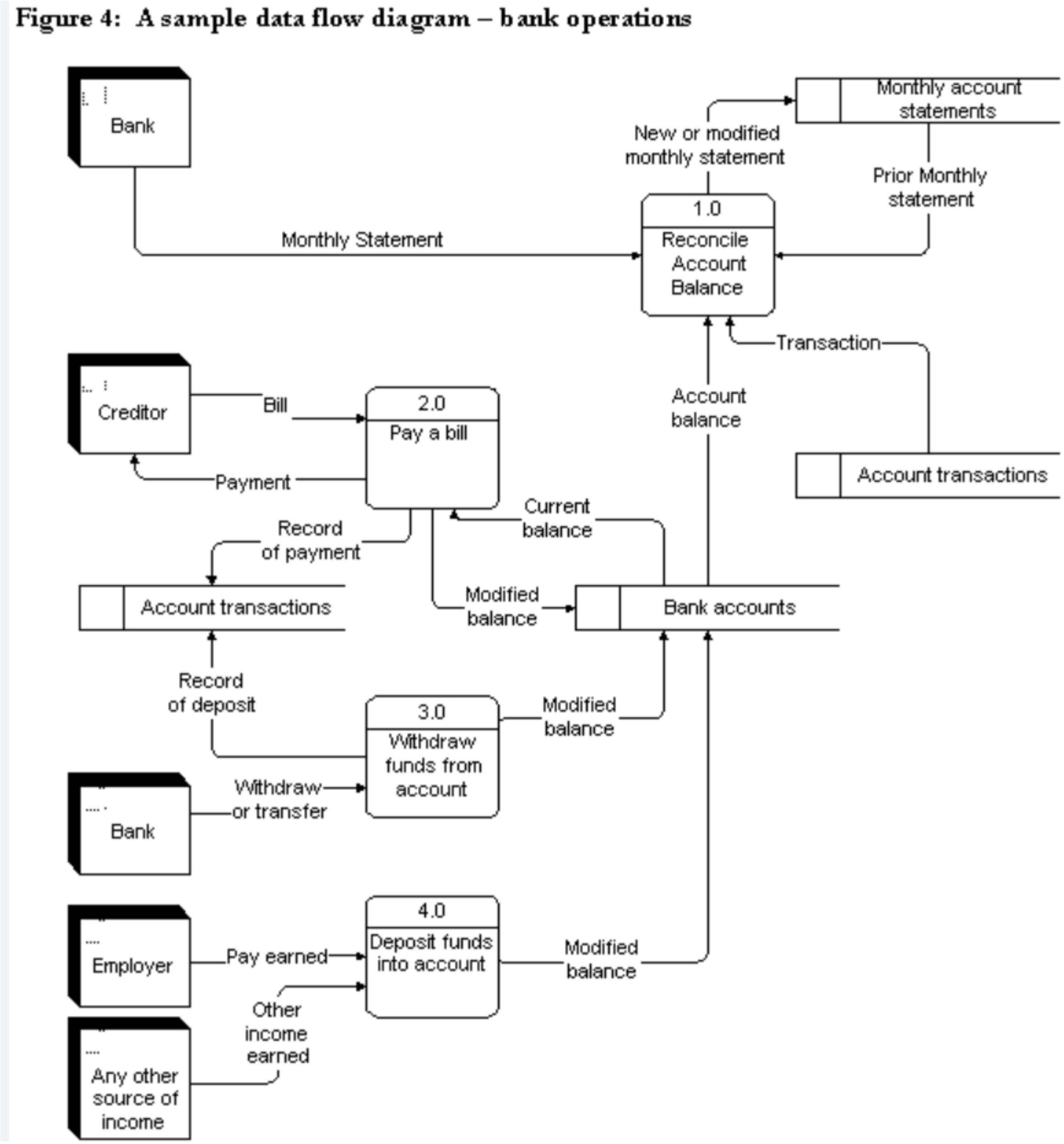

Figure 7 Data Flow Diagram (Source: Babson College)

Another important diagram for understanding databases and the modelling that is required is the Entity Relationship Diagram (ERD). ERDs are essential for understanding the relationships between tables in a database:

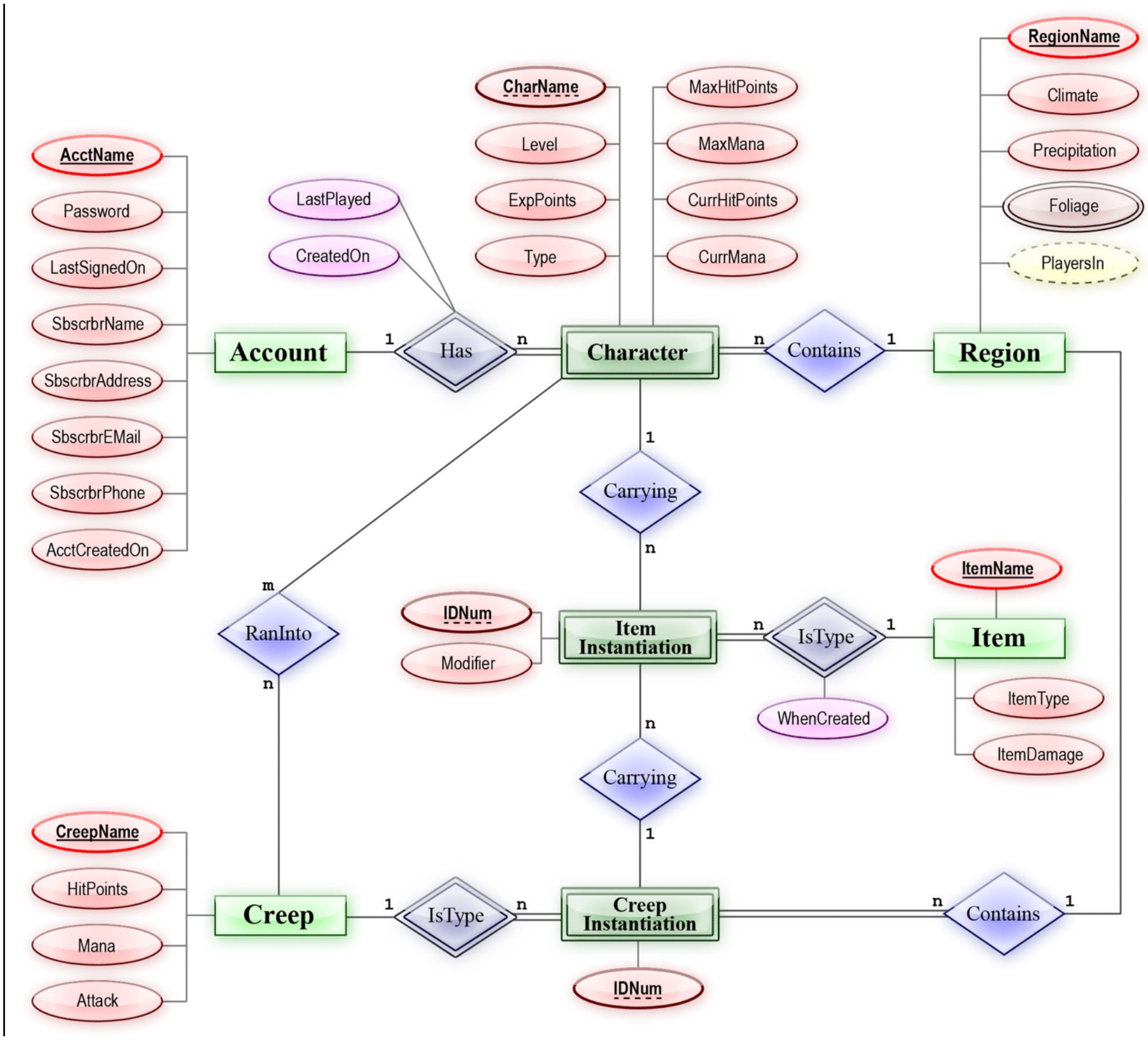

Figure 8 Entity Relationship Diagram (Source: Wikipedia)

Software Design and Architecture

Well-designed software is the goal of software engineering. “The best design for a system is the simplest design that supports all the required features of that system while simultaneously affording the greatest flexibility for change. A simple design in which high-level policies are ignorant of low-level details. Those high-level policies are sequestered and isolated from low-level details such that changes to the low-level details have no impact on the high-level policies." (p. 224 – 225, Martin, Clean Craftsmanship) It's important to keep software design simple to avoid overwhelming complexity. Software design is complex. It is the job of the software engineer to simplify that complexity. "Once complexity has accumulated, it is hard to eliminate, since fixing a single dependency or obscurity will not by itself, make a big difference." (p. 11, Ousterhout) Software engineers need to take a strategic approach to programming. They must focus on the long-term structure of the code and the application. "An abstraction is a simplified view of an entity, which omits unimportant details." (p. 21, Ousterhout)

"When designing modules such as classes and methods, one of the best ways to produce a deep API is to make it general-purpose (general-purpose APIs result in more information hiding)." (p. 39, Ousterhout) "It is more important for a module to have a simple interface than a simple implementation." (p. 61, Ousterhout) An important problem to solve in software design is to either combine functionality in a single class or to separate functionality into more than one class. Functionality should be brought together to eliminate duplicate code. General-purpose code should be separated from code that serves a specific purpose.

When implementing a method, ensure that it does only one thing.

Programming

See Chapter 2 for more information on programming.

Testing and Quality Assurance

Testing and test design are essential to delivering an effective, correct, and efficient software solution. Much time should be devoted to testing to eliminate potential bugs from appearing in the software in production.

Deployment

Deployment is the process of releasing software to a designated environment, whether it is a test environment or production environment. Releasing software to a production environment can be complex, and it is best to streamline the process.

Maintenance

Maintenance involves monitoring software after it has been release to a production environment.

Chapter 10: Programming

Programming

It's essential to properly name variables, methods, and classes to communicate exactly what the intent of the code is. When naming classes, methods, and variables, it's important to use concise and purposeful names. "Good names are a form of documentation: they make code easier to understand." (p. 121, Ousterhout)

It's important to maintain consistency when designing software. Style guidelines provide criteria for developers to write code effectively. "Modern style guides address a range of issues, such as indentation, curly-brace placement, order of declarations, naming, commenting, and restrictions on language features considered dangerous." (p. 143 – 144, Ousterhout)

Make judicious use of white space in code. Blank lines are especially useful in separating blocks of code.

Code should be continuously improved every time programmers work on it.

Debugging

Refactoring

Refactoring code is essential to delivering a well-oiled and functioning software systems. It's important to first get the code to work as expected and then refactor the code to be efficient.

Refactoring helps to eliminate duplicate code throughout the application. "Refactoring is a sequence of small changes that improve the structure of the software without changing its behavior—as proven by passing a comprehensive suite of tests after each change in the sequence." (p. 199, Martin, Clean Craftsmanship)

Source Code Control

Continuous Integration

Continuous integration (CI) is made up of frequent commits of code.

Documentation

It's essential for programmers to document their work through comments in the code, test designs, ReadMe files, and more. However, the code itself should be self-documenting. When developers write comments, those "comments should describe things that aren't obvious from the code." (p. 101, Ousterhout) When writing comments, it is important not to repeat the code.

Unit Tests

Unit tests should provide sufficient and ample coverage of code.

Developers can embrace Test-Driven Design (TDD) and write test cases before developing well-functioning and clean code. TDD improves the design of code and encourages the developer to write de-coupled code.

Acceptance tests are necessary to satisfy the business requirements of developed software.

Code Reviews

“Code reviews provide an opportunity for enforcing conventions and for educating new developers about the conventions. The more nit-picky that code reviewers are, the more quickly everyone on the team will learn the conventions, and the cleaner the code will be.” (p. 145, Ousterhout)

- Questions to ask during a code review

Chapter 11: Object-Oriented Programming

Object-oriented programming takes advantage of modular design to encapsulate complexity.

The pillars of Object-Oriented Programming include the following:

- Abstraction
  - The process of exposing essential features of an entity by hiding the other irrelevant details
  - Reduces complexity and increases efficiency
  - Can have multiple abstractions
- Encapsulation
  - The process of putting data and the operations (functions) that can be performed on that data into a single container called class
  - Any programmer can use this class without knowing how it is implemented
  - Data is insulated and not directly accessible from the outside world
  - Leads to data hiding
- Inheritance
  - The definition of new classes as extensions of existing classes
  - The new class may have some additional properties and functionality
- Polymorphism
  - The ability to take multiple forms and allows the ability to invoke derived class methods through a base class reference during run-time

Chapter 12: Design Patterns

Chapter 13: Database Design

Chapter 14: User Experience

Chapter 15: User Interface Design

Chapter 16: Agile Leadership

The Agile Business Consortium offers the Nine Principles of Agile Leadership. Here they are:

1. Actions speak louder than words

   "Agile leadership is about not only driving and promoting change, it is also about being the change. Those who lead by example and actively engage in their own development, inspire people. This is through action rather than words; as Gandhi said, "Be the change you want to see". Agile Leaders develop themselves to be humble and empathetic by demonstrating virtues such as compassion, kindness and care for their colleagues. Inspiring leaders work on themselves first before working on others."

2. Improved quality of thinking leads to improved outcomes

   "Agile leaders value high quality thinking which will result in meaningful action. Agile leaders view problems from many different angles. They take input from those closest to the problem and this goes some way to ensuring that they are in touch with reality rather than relying solely on electronic information to inform their decision making. This also means allowing thinking time and focusing on the highest priorities at any given time."

3. Organizations improve through effective feedback

   "Receiving feedback can often be perceived as a negative experience, so Agile Leaders lead the way by courageously soliciting meaningful, useful and timely feedback from peers and other colleagues. While requesting feedback is important, agile leaders take time to ensure that they are visibly responding to the suggestions made by their colleagues in order to close the feedback loop. Agile leaders model giving effective feedback that is open, honest and respectful."

4. People require meaning and purpose to make work fulfilling

   "Agile leaders focus on building and sharing a common understanding and purpose. There is a vision of change that is meaningful and applicable to the organization. The work of the agile leader is to be aware of what is in the hearts and minds of their colleagues, and then to unify and align those values into inspired action."

5. Emotion is a foundation to enhanced creativity and innovation

   "Agile leaders inspire others to bring their best selves to their work. They understand that emotion is an important part of the human experience, and when individuals work with their emotions, they achieve more of their potential. Innovation and creativity rely heavily on respect that the agile leader encourages by being accessible, open, honest and transparent whilst expecting the same from others."

6. Leadership lives everywhere in the organization

   "Agile leadership should permeate all aspects of an organization or change initiative. Realizing the leadership potential of all its people helps accelerate the organization's ability to learn and adapt. The work of an agile leader is to develop depth in the organization's leadership capability by providing opportunities for their people to lead. Mentoring tomorrow's leaders in the principles and practices of servant leadership sows the seeds for the agile culture to thrive."

7. Leaders devolve appropriate power and authority

   "Agile leaders recognize that people work best when they are enabled, engaged and energized. Empowering individuals is a necessary skill of the agile leader as they balance the emerging needs and tensions of the organization. Agile leaders recognize that empowerment is not an "all or nothing" concept. Instead, it is a continuum of leadership behavior that responds to the current context for change."

8. Collaborative communities achieve more than individuals

   "Agile leaders build communities based on high trust, respect and meaningful working relationships. Their role is to provide those communities with all that they need to operate efficiently but then to let them function autonomously within their boundaries. The Agile Leader understands that forgiveness, positivity, generosity and gratitude are important parts of a healthy working environment. The healthy functioning of the group together with the preservation of psychological safety, allows the agile leader to encourage learning and development whilst also balancing sustained output and performance for the benefit of the organization."

9. Great ideas can come from anywhere in the organization

   "People who are close to a problem usually have the best ideas about how to solve it. Agile leaders allow themselves to be open to the influence and ideas of others, regardless of their status or position. To this end, the agile leader stops, listens and gives time to really hear the thoughts and ideas for improvement from their colleagues. Even if some ideas are not used, the agile leader encourages a continuous flow of creativity by helping people to understand which ideas were useful and which were not."

Inclusive Leadership

Bill Clinton

Chapter 17: Agile Project Management

Managing legacy systems while maintaining new versions of those systems

Project Management

Well-run software projects meet the following objectives (p.33, McConnell, Professional Software Development):

- Minimal defects
- Maximum user satisfaction
- Minimal response time
- Good maintainability
- Good extendability
- High robustness
- High correctness

- Joining a project in the middle of development

Waterfall Project Management

The Waterfall model of project management involves delivering all of the features of software to the end user at the end of a project. "Waterfall development attempts to minimize the amount of risk in a project by doing extensive planning and documentation prior to initiating construction of the software." (p. 383, Wiegers and Beatty)

Iterative Development

Iterative development aims to deliver software in incremental development, meaning that features are delivered periodically to the end user during the project.

Agile Project Management / Scrum

Agile project management, or Scrum, is a form of iterative project management. It comes from Lean Manufacturing. Agile development involves rapid and frequent delivery of incremental functionality. Agile development is appropriate when requirements inevitably change during a project. Agile project management breaks up the software development process into iterations, or sprints. In each sprint, the requirements are re-prioritized so that the most critical functionality is delivered first. "The goal is to have a body of potentially shippable software at the end of each iteration, even if it constitutes just a small portion of the ultimately desired product." (p. 385, Wiegers and Beatty) Customers are involved continuously during Agile development. Much of the documentation used by developers and QA professionals is contained in user stories. Product backlogs are used to prioritize requirements.

The Agile Manifesto is stated as:

"We are uncovering better ways of developing software by doing it and helping others do it. Through this work, we have come to value:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation

- Customer collaboration over contract negotiation
- Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.” (Agile Manifesto)

User Stories

Waterfall Project Management with Agile Sprints

Some organizations that develop software in large multi-year projects do so in a combined Waterfall/Agile project management approach, where features are delivered incrementally over an expanded period of time.

Appendix A: Attributes of Leaders

1. Leaders take the initiative to do things.

2. Leaders lead by setting an example.

3. Leaders have the courage to change things when it is necessary.

4. Leaders must be willing to ask questions and pursue answers to those questions.

5. Leaders maintain goodwill when it is necessary.

6. Leaders instill in workers the kind of drive and creativity and innovative spirit more commonly found among entrepreneurs.

7. Leaders make an effort to communicate effectively.

8. Leaders work to build trust through honesty and integrity.

9. Leaders who aim for perfection can achieve greatness.

10. Leaders come up with several solutions to each problem and select the best one.

11. Leaders express clearly their expectations of others.

12. Leaders set clear objectives.

Sources

Preparation

The Student Leadership Challenge: Five Practices for Becoming an Exemplary Leader – James M. Kouzes and Barry Posner

Launch Your Career: How ANY Student Can Create Relationships with Professionals and Land the Jobs and Internships They Want – Sean O'Keefe

The First 90 Days: Proven Strategies for Getting Up to Speed Faster and Smarter – Michael D. Watkins

Harvard Business Review – How to Succeed Quickly in a New Role – Rob Cross, Greg Pryor, and David Sylvester - https://hbr.org/2021/11/how-to-succeed-quickly-in-a-new-role - Accessed February 3, 2024

Leadership 101: What Every Leader Needs to Know – John C. Maxwell

Personal Professional Development

Psychology: A Complete Introduction – Sandi Mann

Invitation to Psychology – John P. Houston, Constance Hammen, Amado Padilla, Helen Bee

Harvard Business Review – How to Motivate Employees to Go Beyond Their Jobs – Mark C. Bolino and Anthony C. Klotz - https://hbr.org/2017/09/how-to-motivate-employees-to-go-beyond-their-jobs - Accessed October 4, 2023

Harvard Business Review – Motivating Employees Is Not About Carrots Or Sticks – Lisa Lai - https://hbr.org/2017/06/motivating-employees-is-not-about-carrots-or-sticks - Accessed October 6, 2023

Harvard Business Review – How Company Culture Shapes Employee Motivation – Lindsay McGregor and Neel Doshi - https://hbr.org/2015/11/how-company-culture-shapes-employee-motivation - Accessed December 23, 2023

The Art of Motivational Leadership: A Guide to Becoming a Motivational Leader – Michael P. Tower

Meet Maslow: How Understanding the Priorities of Those Around Us Can Lead to Harmony and Improvement – Landon T. Smith

Good Power: Leading Positive Change in Our Lives, Work, and World – Ginni Rometty

The Ultimate Guide to Power & Influence: Everything You Need to Know – Robert L. Dilenschneider

The Craftsman – Richard Sennett

Clean Craftsmanship: Disciplines, Standards, and Ethics – Bob Martin

Medium.com - Accountability in Software Development – Kent Beck - https://medium.com/@kentbeck_7670/accountability-in-software-development-375d42932813 - Accessed October 8, 2023

Habits and Productivity

Atomic Habits – James Clear

The Power of Habit – Charles Duhigg

Whatever It Takes: Master the Habits to Transform Your Business, Relationships, and Life – Brandon Bornancin

Getting Things Done: The Art of Stress-Free Productivity – David Allen

The 80/20 Principle: The Secret to Achieving More with Less – Richard Koch

Eat That Frog!: 21 Great Ways to Stop Procrastinating and Get More Done in Less Time – Brian Tracy

Feel-Good Productivity: How to Do More of What Matters to You – Ali Abdaal

Problem Solving

Harvard Business Review – The Surprising Power of Questions – Alison Wood Brooks and Leslie K. John - https://hbr.org/2018/05/the-surprising-power-of-questions - Accessed September 29, 2023

Harvard Business Review - Good Leadership Is About Asking Good Questions – John Hagel III – https://hbr.org/2021/01/good-leadership-is-about-asking-good-questions – Accessed August 24, 2023

MIT Sloan School of Management – How Leaders Can Get the Most Out of Asking Questions – Meredith Somers - https://mitsloan.mit.edu/ideas-made-to-matter/how-leaders-can-get-most-out-asking-questions - Accessed September 29, 2023

Leading with Questions: How Leaders Discover Powerful Answers by Knowing How and What to Ask – Michael J. Marquardt

Conceptual Blockbusting: A Guide to Better Ideas – James L. Adams

The Delivery Man: The Art of Turning Ideas Into Products in Silicon Valley – Sebastien L. Taveau, David Birch, and Jean-Louis Gassee

Move Fast and Fix Things: The Trusted Leader's Guide to Solving Problems – Anne Morriss and Frances X. Frei

Harvard Business Review – To Solve a Problem, Reframe It – Julia Binder and Michael D. Watkins - https://hbr.org/2024/01/to-solve-a-tough-problem-reframe-it - Accessed December 23, 2023

HBR on Leadership – Do You Understand the Problem You're Trying to Solve

Boundary Spanning Leadership: Six Practices for Solving Problems, Driving Innovation, and Transforming Organizations – Chris Ernst and Donna Chrobot-Mason

Bulletproof Problem Solving: The One Skill That Changes Everything – Charles Conn and Robert McLean

Collaborative Problem Solving: A Guide to Improving Your Workspace – Chris J. Shannon

An Expert Guide to Problem Solving: With Practical Examples – Aditi Agarwal

Becoming a Technical Leader: An Organic Problem-Solving Approach – Gerald Weinberg

Critical Thinking

The Socratic Method: A Practitioner's Handbook – Ward Farnsworth

The Socratic Way of Questioning: How to Use Socrates' Method to Discover the Truth and Argue Wisely – Thinknetic

Critical Thinking & Logic Mastery: How to Make Smarter Decisions, Conquer Logical Fallacies and Sharpen Your Thinking – Thinknetic

The Critical Thinker: A Guide to Effective Problem Solving, Precision in Decision Making, and Cultivated Mental Discipline – Steven Schuster

HBR Guide to Critical Thinking – Harvard Business Review

Critical Thinking for Managers: Structured Decision-Making and Persuasion in Business – Radu Atanasiu

Engineering

Leadership by Engineers and Scientists: Professional Skills Needed to Succeed in a Changing World – Dennis W. Hess

The Staff Engineer's Path: A Guide for Individual Contributors Navigating Growth – Tanya Reilly

Design Paradigms: Case Histories of Error and Judgment in Engineering – Henry Petroski

TechTarget - Parkinson's Law of Triviality (Bikeshedding) – Amanda Hetler - https://www.techtarget.com/whatis/definition/Parkinsons-law-of-triviality-bikeshedding#:~:text=Bikeshedding%20is%20another%20common%20term,for%20a%20nuclear%20power%20plant

Efficiency and Quality

The Principles of Scientific Management – Frederick Winslow Taylor

Mastering Industrial Engineering: Achieving Efficiency, Innovation, and Excellence – Abdur Rahman

This is Lean: Resolving the Efficiency Paradox – Niklas Modig

Out of the Crisis – W. Edwards Deming

Juran on Quality By Design – J. M. Juran

What Is Total Quality Control?: The Japanese Way – Kaoru Ishikawa

Total Quality Control – Armand V. Feigenbaum

The Continual Improvement Process – Craig Cochran

The Goal: A Process of Ongoing Improvement – Eliyahu M. Goldratt and Jeff Cox

Kaizen: The Japanese Secret to Lasting Change – Sarah Harvey

Kaizen: The Key to Japan's Competitive Success – Masaaki Imai

Six Sigma: A Complete Step-by-Step Guide – The Council for Six Sigma Certification

Lean Thinking: Banish Waste and Create Wealth in Your Corporation – James P. Womack and Daniel T. Jones

Lean Six Sigma: 3 Books in 1: The Complete Guide to Implementing Six Sigma Methodology for Continuous Manufacturing Process Improvement – Jeffrey Ries

Toyota

The Toyota Product Development System: Integrating People, Process and Technology – James M. Morgan and Jeffrey K. Liker

The Toyota Way: 14 Management Principles from the World's Greatest Manufacturer – Jeffrey Liker

The Secret Behind the Success of Toyota: How the Original Chief Engineer System Works to Generate Most of the Product Value and Profit – Takao Sakai

Toyota Production System – Taiichi Ohno

A Study of the Toyota Production System: From an Industrial Engineering Viewpoint (Produce What Is Needed, When It's Needed) – Shigeo Shingo and Andrew P. Dillon

The Machine That Changed the World – James P. Womack, Daniel T. Jones, and Daniel Roos

Software Engineering

Software Engineering for Absolute Beginners: Your Guide to Creating Software Products – Nico Loubser

Engineers Survival Guide – Merih Taze

Software Engineering – Ian Sommerville

AWS – What Is SDLC (Software Development Lifecycle)? - https://aws.amazon.com/what-is/sdlc/#:~:text=The%20software%20development%20lifecycle%20(SDLC,expectations%20during%20production%20and%20beyond.

Software Requirements – Karl Wiegers and Joy Beatty

More About Software Requirements – Karl Wiegers

Use Cases – Usability.gov - https://www.usability.gov/how-to-and-tools/methods/use-cases.html

A Beginner's Guide to Data Flow Diagrams – Hubspot – Clifford Chi - https://blog.hubspot.com/marketing/data-flow-diagram

A Philosophy of Software Design – John Ousterhout

Clean Architecture: A Craftsman's Guide to Software Structure and Design – Bob Martin

Microsoft .NET: Architecting Applications for the Enterprise – Dino Esposito and Andrea Saltarello

Software Architecture in Practice – Len Bass, Paul Clements, and Rick Kazman

Software Engineering Economics – Barry W. Boehm

Programming

Code Complete – Steve McConnell

Clean Code – Bob Martin

Clean Craftsmanship: Disciplines, Standards, and Ethics – Bob Martin

Coders at Work – Peter Seibel

Rapid Development – Steve McConnell

Refactoring: Improving the Design of Existing Code – Martin Fowler

The Nature of Software Development: Keep It Simple, Make It Valuable, Build It Piece by Piece – Ron Jeffries

The Pragmatic Programmer: Your Journey to Mastery – David Thomas and Andrew Hunt

Programming Pearls – Jon Bentley

Soft Skills: The Software Developer’s Life Manual – John Sonmez

Object-Oriented Programming

The Object-Oriented Thought Process – Matt Weisfeld

Head First Object-Oriented Analysis & Design – Brett D. McLaughlin, Gary Pollice, and David West

Object-Oriented Analysis – Peter Coad and Edward Yourdon

Object-Oriented Design – Peter Coad and Edward Yourdon

Designing Object-Oriented Software – Rebecca Wirfs-Brock, Brian Wilkerson, and Lauren Wiener

Object Design: Roles, Responsibilities, and Collaborations – Rebecca Wirfs-Brock and Alan McKean

Object-Oriented Programming – Peter Coad and Jill Nicola

Object-Oriented Software Construction – Bertrand Meyer

Design Patterns

Adaptive Code via C#: Agile Coding with Design Patterns and SOLID Principles – Gary McLean Hall

Head First Design Patterns: A Brain-Friendly Guide – Eric Freeman and Elisabeth Robson

Dependency Injection in .NET – Mark Seemann

Design Patterns: Elements of Reusable Object-Oriented Software – Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides

Professional ASP.NET Design Patterns – Scott Millett

Patterns, Principles, and Practices of Domain-Driven Design – Scott Millett and Nick Tune

Enterprise Integration Patterns – Gregor Hohpe and Bobby Woolf

The Art of Unit Testing – Roy Osherove

Software Testing Techniques: Finding the Defects that Matter – Scott Loveland, Geoffrey Miller, Richard Prewitt, and Michael Shannon

Database Design

A First Course in Database Systems – Jeffrey D. Ullman and Jennifer Widom

SQL Fundamentals – John J. Patrick

Microsoft SQL Server 2008 T-SQL Fundamentals – Itzik Ben-Gan

Programming SQL Server 2008 – Leonard Lobel, Andrew J. Brust, and Stephen Forte

Inside Microsoft SQL Server 2008: T-SQL Programming – Itzik Ben-Gan

SQL Server 2008 Query Performance Tuning Distilled – Grant Fritchey and Sajal Dam

Inside SQL Server 2008: T-SQL Querying – Itzik Ben-Gan

Microsoft SQL Server 2008 Internals – Kalen Delaney

User Experience

The Elements of User Experience – Jesse James Garrett

The Design of Everyday Things – Don Norman

User Interface Design

Human-Computer Interaction – Alan Dix, Janet Finlay, Greogry D. Abowd, and Russell Beale

Don't Make Me Think: A Common Sense Approach to Web and Mobile Usability – Steve Krug

A Guide to Usability: Human Factors in Computing – Jenny Preece

Agile Leadership

Theory of Constraints – Eliyahu M. Goldratt

Agile Business Consortium – The Nine Principles of Agile Leadership - https://www.agilebusiness.org/resource/the-nine-principles-of-agile-leadership.html

Strategic Doing: Ten Skills for Agile Leadership – Edward Morrison, Scott Hutcheson, Elizabeth Nilsen, Janyce Fadden, and Nancy Franklin

Doing Agile Right: Transformation Without Chaos – Darrell Rigby, Sarah Elk, and Steve Berez

The Rise of the Agile Leader: Can You Make the Shift – Chuck Mollor

The Professional Agile Leader: The Leader's Journey Toward Growing Mature Agile Teams and Organizations – Ron Eringa, Kurt Bittner, and Laurens Bonnema

Agile Leadership Toolkit: Learning to Thrive with Self-Managing Teams – Peter Koning

The Epic Guide to Agile: More Business Value on a Predictable Schedule with Scrum – Dave Todaro

Inclusive Leadership: The Definitive Guide to Developing and Executing an Impactful Diversity and Inclusion Strategy – Charlotte Sweeney and Fleur Bothwick

The Inclusive Leadership Journal: A 90-Day Journey to Becoming a More Inclusive People Leader – Adriele Parker

LinkedIn – Inclusive Leadership for Agile Teams – Rene Carayol - https://www.linkedin.com/pulse/inclusive-leadership-agile-teams-ren%C3%A9-carayol-mbe/ - Accessed September 2, 2024

Harvard Business Review – Why Inclusive Leaders Are Good for Organizations, and How to Become One – Juliet Bourke and Andrea Titus - https://hbr.org/2019/03/why-inclusive-leaders-are-good-for-organizations-and-how-to-become-one - Accessed September 30, 2023

Heidrick & Struggles – Six Actions Leaders Can Take to Nurture Inclusive Teams – Alice Breeden and TA Mitchell - https://www.heidrick.com/en/insights/diversity-inclusion/six-actions-leaders-can-take-to-nurture-inclusive-teams - Accessed September 15, 2024

My Life – Bill Clinton

The Survivor: Bill Clinton in the White House – John F. Harris

First in His Class: A Biography of Bill Clinton – David Maraniss

The Agenda: Inside the Clinton White House – Bob Woodward

Bill Clinton: Mastering the Presidency – Nigel Hamilton

Bill Clinton: An American Journey – Nigel Hamilton

Bill Clinton: New Gilded Age President – Patrick J. Maney

Agile Project Management

HBR on Leadership – NASA's Former Head of Science on What It Takes to Manage Complex, High-Risk Projects

Harvard Business Review – Why Big Projects Fail – and How to Give Yours a Better Chance of Success – Te Wu and Ram B. Misra - https://hbr.org/2023/11/why-big-projects-fail-and-how-to-give-yours-a-better-chance-of-success - Accessed November 3, 2023

Harvard Business Review – How Project Managers Can Better Navigate Setbacks – Amy Shoenthal - https://hbr.org/2023/10/how-project-managers-can-better-navigate-setbacks?ab=HP-hero-latest-text-2 – Accessed October 30, 2023

Harvard Business Review – Keep Your Team Motivated When a Project Goes Off the Rails – Rebecca Zucker - https://hbr.org/2023/10/keep-your-team-motivated-when-a-project-goes-off-the-rails - Accessed November 5, 2023

Harvard Business Review – Project Managers, Focus on Outcomes – Not Deliverables – Andrea Belk Olson - https://hbr.org/2023/11/project-managers-focus-on-outcomes-not-deliverables - Accessed November 3, 2023

Harvard Business Review – Project Managers Should Think Like Startup Founders – Ron Ashkenas - https://hbr.org/2023/11/project-managers-should-think-like-startup-founders - Accessed November 5, 2023

Harvard Business Review – What the Next Generation of Project Management Will Look Like – Rachel Longhurst and Woojin Choi - https://hbr.org/2023/11/what-the-next-generation-of-project-management-will-look-like - Accessed November 6, 2023

Software Project Survival Guide – Steve McConnell

The Mythical Man-Month – Frederick P. Brooks

Working Effectively with Legacy Code – Michael C. Feathers

Agile Manifesto - https://agilemanifesto.org/

Harvard Business Review – Planning Doesn't Have to Be the Enemy of Agile – Alessandro Di Fiore - https://hbr.org/2018/09/planning-doesnt-have-to-be-the-enemy-of-agile - Accessed September 18, 2023

Agile Project Management with Scrum – Ken Schwaber

Succeeding With Agile – Mike Cohn

Agile Estimating and Planning – Mike Cohn

Agile Testing – Lisa Crispin and Janet Gregory

User Stories Applied – Mike Cohn

Additional Sources

Professional Software Development – Steve McConnell

Leadership: Theory and Practice – Peter G. Northouse

Leadership: Theory, Application, & Skill Development – Robert N. Lussier and Christopher F. Achua