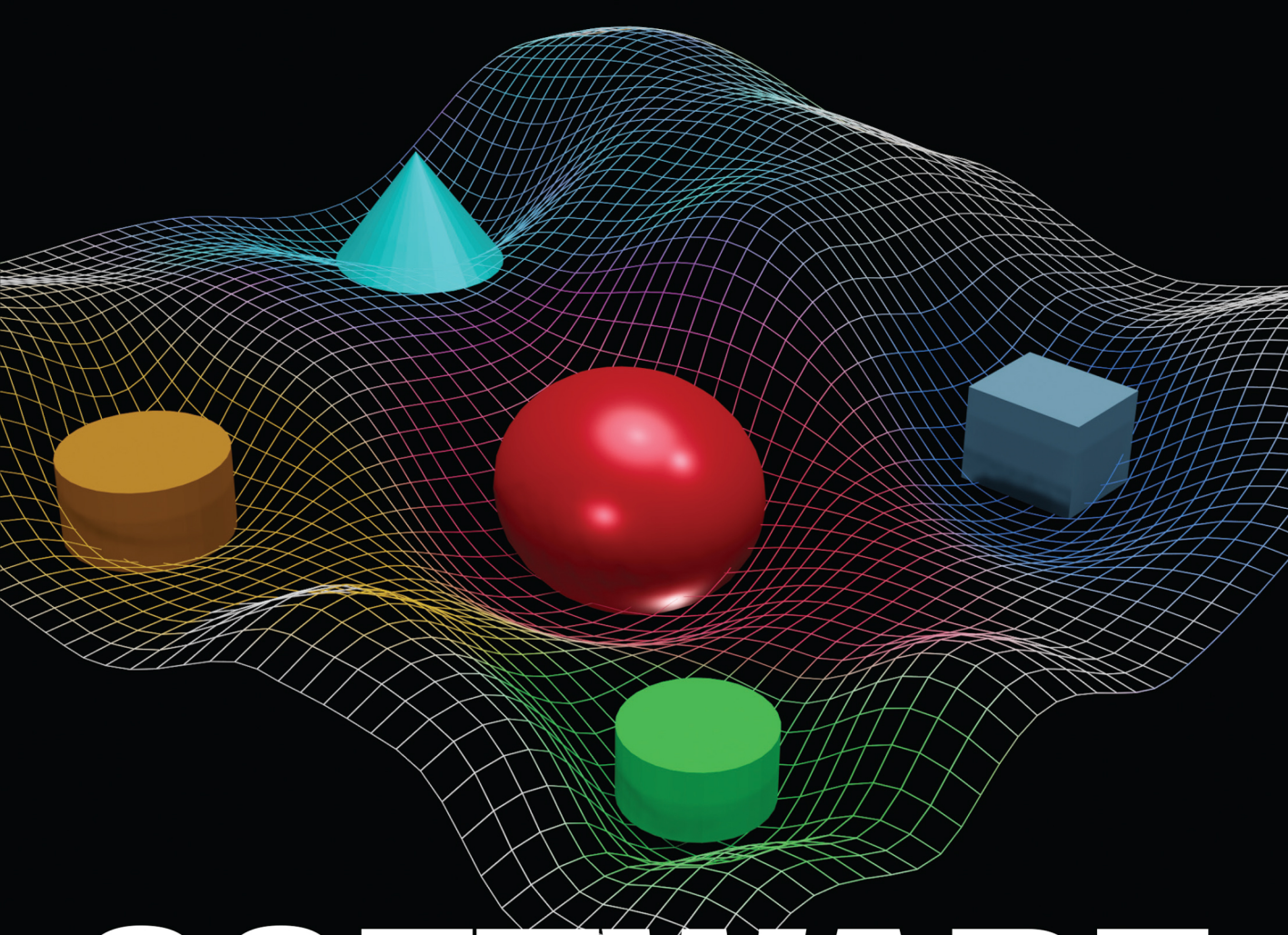
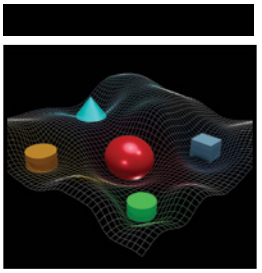


SOFTWARE ARCHITECT

MICHAEL BELL

WILEY

Software Architect



Software Architect

Michael Bell

WILEY

Published by John Wiley & Sons, Inc., Hoboken, New Jersey.
Published simultaneously in Canada and the United Kingdom.

ISBN: 978-1-119-82097-0
ISBN: 978-1-119-82098-7 (ebk.)
ISBN: 978-1-119-82099-4 (ebk.)

Library of Congress Control Number: 2022951780

Cover image: Courtesy of Michael Bell
Cover design: Wiley

About the Author

Michael Bell is an American novelist, artist, producer, and enterprise solution architect, chiefly recognized for developing the Incremental Software Architecture methodology (ISAM), Service-Oriented Modeling Framework (SOMF), Cloud Computing Modeling Notation (CCMN), and the Multidimensional Software Architecture Construction (MSAC). His innovative research and publications in the fields of software architecture, service-oriented architecture, microservices, artificial intelligence (AI), cloud computing, and big data are recognized internationally for their contribution to the software design and development communities. He has consulted for organizations including J.P. Morgan Chase, Citibank, Bank One, UBS-Paine Webber, American Express, AIG, and the U.S. government. He is the best-selling author of software architecture books, and he offers a variety of enterprise integration solutions for back-end and customer-facing systems.

About the Technical Editor

Paul C. Martello is a technical writer with more than 18 years of experience in IT. Before becoming a technical writer, he held roles ranging from elementary and business teacher in New York to history teacher for Fairfax County Public Schools in Virginia. In 2005, Paul was selected to participate in the Relief International Schools Online Teacher Exchange in Bangladesh, integrating technology in schools. He loves watching his favorite football team, the Buffalo Bills, traveling, and spending time with his family in Bristow, Virginia.

About the Contributing Editors

Noreen O'Brien has spent the better part of her life writing, editing, and creating through various mediums. From her start as a reporter to the freelance editing work she does today, she has been instrumental in the production of a multitude of papers, dissertations, and documents. She lives in Richmond, Virginia, with her son, Liam.

Monica Gagnier is an experienced editor who has worked with Michael Bell on previous books. A graduate of Syracuse University's S.I. Newhouse School of Public Communications, Gagnier is a seasoned financial journalist who has worked at such publications as BusinessWeek (now Bloomberg Businessweek), the New York Post, and the Albuquerque Journal. She lives in Santa Fe, N.M.

Contents at a Glance

Contents

Introduction: Software Architect, Who Are You?

As a software architect you've embarked on a career journey in an uncharted and unpredictable territory with no guarantee of successful technological solutions. You are employed as a software architect to participate in a corporate business, technological, and social experiment whose chief thrust is to manufacture software products deployed to virtual environments. It's also arduous to foretell the business performance quality and stability after deploying and integrating software implementations in computing ecosystems.

By no means is this a bleak portrayal of a software architecture career. On the contrary, the uncertainty of your contribution to enterprise solutions only opens the doors to business development and transformation opportunities, technological modernization, and career improvement and growth. Furthermore, your hard work and dedication can be achieved through the power of creativity, imagination, and persistence. Once you are resolved to pursue a software architecture career, or are already a devoted practitioner, you're destined for a highly successful journey.

The following sections draw a picture of an *ideal software architect* whose capability to solve organizational problems is beyond imagination. This profile represents a well-rounded software architect with close-to-perfect professional talents that organizations would most certainly employ if the need existed. However, do not fret or be discouraged. We strive to possess these outlined qualities to make a difference in people's lives by promoting business culture, strategies, mission, and vision.

Figure I.1 illustrates the ideal software architect's attributes: career-oriented, innate traits-driven, strategy-driven, culture promoter, integration-driven, leadership-oriented, solution-driven, domain-driven, and social-driven.

So, ideal software architect, who are you?

Figure I.1: An Ideal Software Architect Profile

You Promote Institutional Culture

You're hired as a software architect to inspire change, stir up enthusiasm for innovation, stimulate new ideas, affect organizational strategies, combat business and technological stagnation, and make a big difference in people's lives.

Become an Agent of Cultural Transformation

You are offered a key position to participate in transforming the *old* into the *new*. The former refers to outdated business concepts, traditional ways of doing business, archaic methods of developing software products, and waning technological solutions. The "new," on the other hand, pertains to modern technologies, creative and practical applications and systems, innovative end-to-end software architecture methodologies and life cycles, and partnerships that promote organizational dialogue to secure the business.

By partaking in such ambitious organizational metamorphosis, you're the de facto *institutional agent of cultural transformation*. You are actively engaged in a social and technological experiment that touches lives and instills change in people's behavior. This multifaceted cultural change manifests in how people communicate, interface with applications and systems, form relationships and partnerships, run their daily lives, and manage their careers.

So, how do software architects promote organizational culture? The arsenal of tools and utilities employed to impact the environment profoundly is vast.

Furthermore, the sky is the limit for technological evolution and innovation. The business and technological solutions you're being asked to provide drive the establishment of organizational *policies*, *best practices*, and *standards*. These rules and procedures you're advocating for promote institutional norms of behavior, foster business alliances, and forge new codes of cultural conduct.

Contribute, Do Not Follow

However, the cultural change that you're promoting does not touch only individuals. You are employed to harness the power of your talents and creativity to form a new generation of ideas and find shared values reinforced by members of your organization inspired by your innovative visions. In reality, you are a benefactor at heart, not a follower. Any organizational solution you offer contributes to the institutional knowledge base and the collective memory of your followers, who are ultimately employed to solve enterprise problems.

Further Reading

Although the topic of promoting organizational culture is discussed throughout the book, refer to these chapters to learn about the specific methods that software architects can leverage to impact institutional culture:

- Chapter 3, "Career Planning for Software Architects: A Winning Strategy" depicts four career-driven perspectives that can impact organizational culture: social-driven, technology-driven, management-driven, and strategy-driven.
- Chapter 4, "Self-Assessment for Software Architects" offers a self-scoring questionnaire that contains queries about promoting organizational culture methods.

You're an Astute Strategist

Your strategic mindset is the key to the success of your software architecture career. No matter which software architecture scope of solutions you pursue, application or enterprise level, focus on the big picture. You're a generalist by nature. Never rush into details to develop effective solutions. Having a bird's-eye view is what makes you an all-around type of person.

You're also a gifted tactician who incessantly occupies your mind with long-term and sustainable solutions to remediate business problems. The prospect of business prosperity and technological continuity motivates you to carve out complicated schedules, road maps, and product development timetables.

No matter the magnitude of your work, your strategic outlook is driven by a thorough study of business and technological events that occur on the ground. Then, by connecting the dots, you deliver superior software architecture artifacts. In this context, *connecting the dots* pertains to aggregating and utilizing all possible organizational resources, such as subject-matter experts, data, utilities, and facilities, to derive the best possible software and environment implementations.

Adopt an Effective Outside-In Strategy to Deliver Synthesized Software Architecture Solutions

You're an outside-in software architecture strategist attuned to market and industry trends, quality of organizational services, and, most important, customer imperatives. In addition, you are acquainted with advanced product development life-cycle methodologies and often follow business market developments and innovations, valuable knowledge that drives your methodological approach to meeting client requirements. Satisfying these imperatives begins with an effective business discovery and analysis process that leads to software architecture solutions.

Do not be constrained by existing technological limitations. If current organizational technologies tend to narrow the scope of your vision, you must drive change, modernization, and initiatives aligned with your software architecture vision and mission. Furthermore, you drive business and technological transformation through creativity, curiosity, and modernity synthesis. Finally, never deprive yourself of the freedom of imagination when proposing innovative software architecture implementations.

Align Software Architecture Strategies with Organizational Imperatives

As an astute software architecture strategist, you know that your technological vision and mission must align with business strategies. Remember, you're not operating in a vacuum. Your software architecture solutions, therefore, ought to promote business agendas, foster business growth, and ensure business stability and continuity.

However, aligning software architecture strategies with business vision and mission would not promote satisfactory technological solutions. Business cooperation and coordination are indeed primary and compulsory goals for software architects. Their duties, however, must go beyond business imperatives. There are accessorial software architecture strategy alignment necessities to drive a comprehensive enterprise technological balance.

Thus, software architecture strategies must also be aligned with existing deployment environments, supporting infrastructure, development platforms,

data and message exchange mechanisms, architecture styles, design patterns, and integration patterns. Again, promote transformation initiatives to satisfy software architecture vision and mission if software architecture strategies cannot align with existing technologies and environments.

Figure I.2 illustrates a software architecture strategy alignment priority example chart that outlines alignment opportunities with business, technologies, environment, and infrastructure.

Figure I.2: Software Architecture Alignment Priorities

Further Reading

The topic of software architecture strategy alignment with business strategy, vision, and mission to propel technological initiatives across the organization is discussed largely in Chapter 2, "Types of Software Architects." It introduces three business needs for software architecture to foster organizational transformation and modernization: strategic collaboration, technological mediation, and technological implementation

You're a Gifted Leader

You're a leader, not necessarily a manager. You possess noteworthy interpersonal traits. You're a person of integrity who instills trust in your co-workers, managers, corporate executives, and partners. You promote institutional social harmony to foster consensus on software architecture strategies, technologies, best practices, standards, and policies. You're trusting and trustworthy because you have a positive perspective of humankind.

Your natural leadership traits inspire followers. These devoted fans respect your perspectives and are committed to collaborating with you on software architecture projects and business initiatives. As a gifted technological leader and team player, you prefer to collaborate with others. You encourage diversity of

ideas and solutions by fostering the collective creativity of enthusiastic technologists. You never impose your views on others—in contrast, you're an advisor, a mentor who offers viable guidance to those who seek professional direction.

TIP **Remember, you're a leader. You're not a manager or administrator who signs timesheets and reprimands staff for wrongdoing.**

Tolerate Errors and Stay Open to Technological Experiences

Your innate problem-solving and decision-making skills paint a realistic view of your organization's business and technological contribution. In other words, nothing is perfect! You understand the difficulties and constraints of any proposed software architecture solution. And you're aware of the impact your technical recommendations have on your organization. You're wise to understand that ill-designed applications and systems can cause operational chaos, disrupt business continuity, negatively impact productivity, and harm your company's bottom line.

With all these potential risks to the enterprise, you're still a natural optimist and idealist, a risk-taker willing to surrender short-term gains in favor of strategic long-term technological success. These traits define a person who tolerates design errors, software implementation mishaps, and software deployment and integration flaws. In reality, you're not afraid of failure. In your mind, the design experiment journey you're willing to embark on can only promote successful technological modernization.

Build a Circle of Trustful Followers by Uplifting Their Spirits

As a software architect, you must inspire others and galvanize positive energy among your co-workers and work teams. You're here to foster creativity—the failure of imagination is not an option. You are here to usher intelligent followers who trust your software architecture judgment and good taste, and who are not afraid of making design mistakes or expressing silly opinions.

TIP **Remember, you're an experimentalist whose leadership traits galvanize enthusiasm for collaborative teamwork to offer superb technological solutions for sustaining and accelerating business success.**

Further Reading

Take the self-evaluation questionnaire provided in Chapter 4, "Self-Assessment for Software Architects," to find out if you possess the proper software architecture

leadership talents that can galvanize enthusiasm for business innovation and technological modernization.

You're an Instrumental Solution Provider

At heart, you are a solution provider. Deep inside you, there is a veiled desire to mitigate risks, resolve social conflicts, and provide guidance to tackle organizational challenges. You always rise to the occasion to remediate business shortfalls. Furthermore, you go the extra mile to seek pragmatic technological solutions.

Promote Business Growth Through Modern Technological Solutions

You are committed to implementing potent strategic foundations for sustainable and viable business growth through technological modernization. You're a risk-taker and venture to support business transformation by leveraging the best-of-breed technical capabilities. Furthermore, you believe that technology is not just a mechanical mean for implementing temporary solutions or for offering Band-Aid remedies that do not withstand time. Simply put, you're a solution provider with technological, strategic agendas that tolerate occasional failures to achieve novel goals.

Provide Solutions Within the Boundaries of Your Software Architecture Expertise

As a software architect, you focus on design solutions—software-oriented remedies, *not hardware*. This is because you understand the boundaries of your occupation. You're aware that the solutions you provide are within the margins of software architecture practices—the field in which you excel. You may collaborate with co-workers specializing in physical infrastructure, hardware servers, and network devices. However, your chief responsibility is to design applications, services, systems, and deployment environments within your software architecture expertise.

Know the boundaries of your responsibilities. Be aware of your software architecture level of contribution. You're wise to understand that the reach of your technical solutions depends on the boundaries of your position. Namely, the job you're holding as a software architect has restricted outreaching responsibilities. This is not because you cannot accomplish tasks beyond your job description. It's simply due to the software architecture duties you're commissioned to pursue.

Understand the Scope of Your Technological Solutions

So, what would be the scope of your technological solutions?

Nowadays, common organizational practice calls for founding a hierarchical structure of software architecture roles. They are established to address three different levels of solutions. Affiliated with the top layer of a pyramid, enterprise software architects and their technological solutions must meet enterprise-level business imperatives. Then, solutions architects are related to the second layer, just beneath the enterprise architects' level. They are commissioned not only to promote enterprise software architecture strategies, but also to oversee application-level design initiatives. Finally, application architecture roles are the nucleus of any organizational software design initiative. They are positioned at the very bottom of the structural employment hierarchy, assigned to offer solutions for the narrowest range of problems. Figure I.3 illustrates the hierarchical structure of software architecture roles and their solution scope and dependencies within an enterprise.

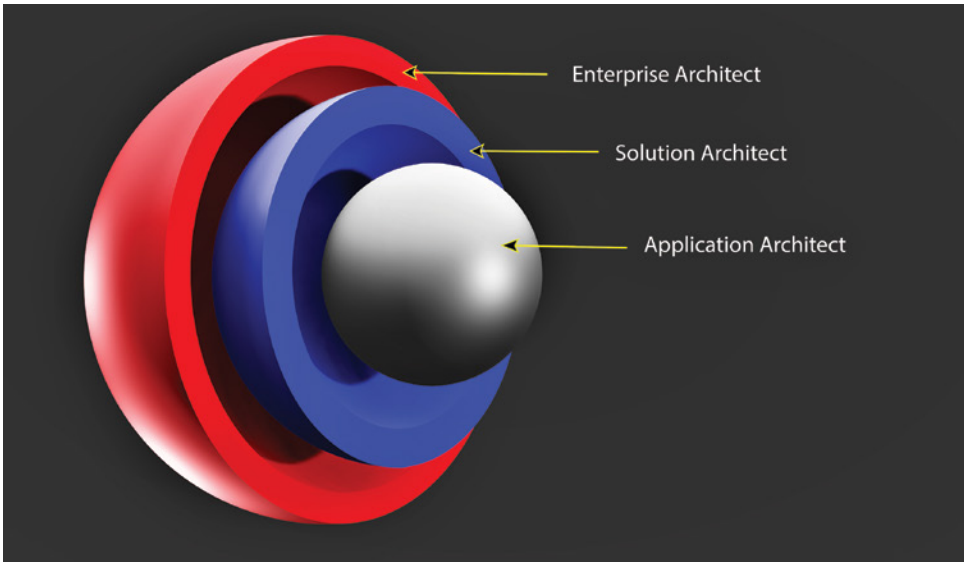


Figure I.3: Software Architecture Roles And Their Organizational Solution Scope

Further Reading

To learn more about how to scope technological solutions and set boundaries for your professional expertise visit these two chapters:

- Chapter 1, "Software Architect Capability Model," discusses a method to help scoping technological solutions and setting boundaries to a professional occupation by creating a capability model with five driving sections:

specifications, architecture practices, architecture disciplines, architecture deliverables, and quantification of skill competencies.

- Chapter 2, "Types of Software Architects," elaborates on two types of software architect roles: leading software architects and domain software architects. Each of these roles are commissioned to focus on specific solution scopes.

You're an Integrator Par Excellence

Integration duties are the bedrock foundation of your occupation. It's an integral part of your professional daily practice. No matter what level of software architecture contribution to the enterprise you provide, you're well aware that integration is a compulsory responsibility that you cannot avoid. It's a software design capability you possess, leverage, and exhibit to satisfy a broad range of business and technological imperatives. Furthermore, integration is a technological, social, and business competence you consistently demonstrate to provide large-scale business remedies. And it's a software architecture aptitude you employ to aggregate solutions mutually provided by a community of software implementations.

Connect the Dots

You're dubbed a "software integrator" because every design scheme you devise proves effective partnerships and communications between software implementations. Any design blueprint you provide presents logical views of interaction and collaboration between applications, services, and systems. And it's starkly apparent that any software architecture environment you design maintains a pragmatic alliance between distributed software assets.

You do not take the term *connecting the dots* lightly in regard to software integration. Namely, you do not sneeze at opportunities to utilize diverse sources of information, combine people's ideas, and aggregate technological fountains of knowledge to devise powerful software architecture integration solutions. In essence, you're wise to connect the dots to foster software reuse and optimize the redundancy of business functionality.

Integrate Software in a Three-Dimensional Software Architecture Environment

As a software architect, you're keenly aware that integration is not only about connecting the dots and not merely about enabling software entities to talk to

each other and exchange information. Indeed, these are vital software architecture tasks that ensure business continuity, ensuing viable organizational solutions.

However, you're also mindful that software implementations do not operate in a vacuum and are deployed to a topological, geometrical, and three-dimensional landscape that offers them adequate architectural conditions to survive in. In Chapter 6, "Software Architecture Environment Construction," and Chapter 7, "Structural Construction of Software Implementations in Multidimensional Environments," this ecosystem is labeled the *software architecture environment*. As illustrated in Figure I.4, this landscape constantly undergoes structural deformation due to the behavior of the hosted software entities.

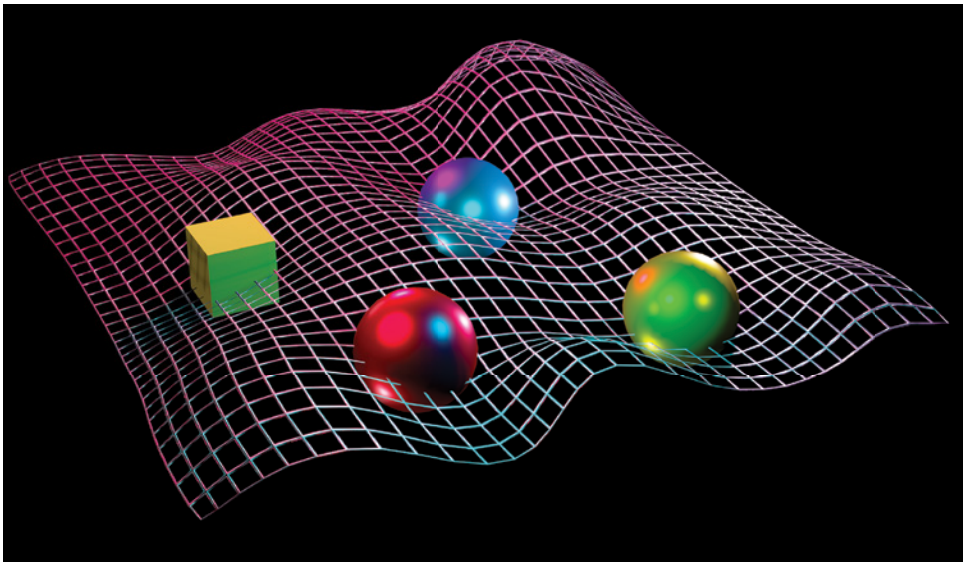


Figure I.4: Structural Deformation Of A Software Architecture Environment

Mitigate Risks in a Quantum Software Architecture Ecosystem

Your design outcomes consistently demonstrate a compromise between radical software architecture solutions. These negotiated solutions between extreme design approaches contribute vastly to the mitigation of the risks of an unpredictable deployment environment that can negatively impact business execution. Also, you're compelled to adhere to integration best practices, standards, and policies to foster a balanced software architecture environment. Your instrumental integration talents promote a sensible environment balance to minimize the erratic behavior of software. And your surpassing software integration capabilities alleviate the risks of business interruptions.

Further Reading

The book's Part 3, "Software Architecture Toolbox," represents the Multidimensional Software Architecture Construction (MSAC) methodology. This design approach offers use cases, best practices, and construction laws for software implementations and their affiliated environments:

- Chapter 6, "Software Architecture Environment Construction," is all about integration of software in a multidimensional software architecture environment hosted in production.
- Chapter 7, "Structural Construction of Software Implementations in Multidimensional Environments," represent the 3D structural composition of software entities that are deployed and integrated in a software architecture space.

You're Domain-Driven

You're well prepared to tackle business and technological problems by employing your software architecture talents. There is nothing that can swerve your focus from fulfilling your goals. Furthermore, your uncompromising devotion to offering effective and sustainable software architecture solutions is immeasurable to your organization. Your steadfast resolve to tackle business challenges is attributed to your laser-beam focus on critical problems while avoiding personal agendas and evading trivial issues.

Simply put, the secret of your unwavering commitment to providing best-of-breed software architecture solutions is rooted in your ability to concentrate on what matters. More specifically, your solutions align with corporate business and technological strategies; software architecture vision and mission; leadership directives; and institutional best practices, standards, and policies.

TIP **In a nutshell, you're a domain-driven software architect familiar with the business environment, the industry, the customers, and the supporting technology.**

Align the Orbit of Your Software Architecture Solutions with Organizational Domains

The alignment of software architecture solutions with business imperatives characteristically yields robust technological solutions. In this context, *business imperatives* refers to different types of requirements. As a pragmatic software architect you can tailor technological solutions to specific business needs. More

explicitly, your solutions to business problems may be affiliated with a specific business sector, business industry, business product, business portfolio, line of business, or business division. These particular business domains drive the technical remedies you propose.

However, business needs do not always drive the domain alignment process. Equally significant is the alignment *of software architecture strategies with business strategies*. The chief reason is that business strategies are the empirical driving forces in the enterprise. Therefore, technological solutions should foster and support long-term business plans, business models, business vision and mission, and business policies.

Moreover, from a domain alignment perspective, you're most certainly aware that the existing technological capabilities of your organization (such as infrastructure, platforms, and networks) must support software architecture solutions. In some cases, the existing technological capacity may not be advanced enough to deliver your proposed design. Therefore, promote technological modernization and transformation initiatives to improve the alignment with your architectural vision and mission.

Delineate the Scope of Your Software Architecture Solutions

Your devotion to providing software architecture solutions to specific organizational imperatives accelerates time-to-market and ensures business and technological continuity. Pinpointing the sources of business obstacles, devising feasible solutions, and mitigating domain-related issues are prescriptions for software architecture success.

TIP **By accomplishing this, you're essentially accredited as a domain-driven software architect who is business-driven, strategy-driven, technology-driven, solution-driven, and leadership-driven. Leverage these capabilities to respond to business and technological necessities.**

Again, aligning your technological solutions with organizational domains promotes pragmatic software architecture. Therefore, it's highly advisable to create a solution-focused domain diagram similar to the one shown in Figure I.5. Such a depiction will demonstrate the various opportunities for software architecture success. Focus on your organizational domains that require attention. Leverage your leadership talents to focus on particular business and technological problems. Finally, focus only on domain challenges that require solutions.

Figure I.5: Domain-Driven Software Architecture Solution Scope

Further Reading

As a domain-driven software architect whose duty is to deliver a balanced software architecture, focus on the software architecture construction life cycles, governing laws, and best practices covered in Chapter 6, "Software Architecture Environment Construction" and Chapter 7, "Structural Construction of Software Implementations in Multidimensional Environments."

Furthermore, to foster a software architecture environment equilibrium, employ your *domain-driven design skills* to meet business, strategy, technology, solution, and leadership imperatives.

You're Socially Driven

As a software architect, you're mindful that social collaboration and partnership with co-workers, industry alliances, customers, and stakeholders yields compelling technological solutions. Technical solutions have never been successful without teamwork and cooperation with subject-matter experts (SMEs). Individuals promoting personal agendas can never deliver substantial software architecture strategies.

In conclusion, software architects should fulfill their duties through the power of social intelligence. Moreover, architects who snub social skills to better accomplish the tasks they were hired for often find that their software architecture solutions ultimately fail to live up to organizational expectations. Simply put,

the respectful and productive interrelationship between technologists and business leaders always demonstrates social capabilities that deliver perceptive technological solutions.

Leverage the Contribution of Social Intelligence to Your Software Architecture Career

In this context, *social intelligence* pertains to your ability to understand yourself, your needs, and your limitations. However, it's not only about your imperatives or boundaries. This self-acumen is about the capability to know others, the aptitude to understand the environment, and the faculty to develop trustful and sustainable partnerships in the workplace.

Have a look at the chief social intelligence tokens presented in Figure I.6. These represent software architecture social intelligence capabilities leveraged to drive powerful business and technology transformation solutions: *agility*, *adaptability*, and *awareness*.

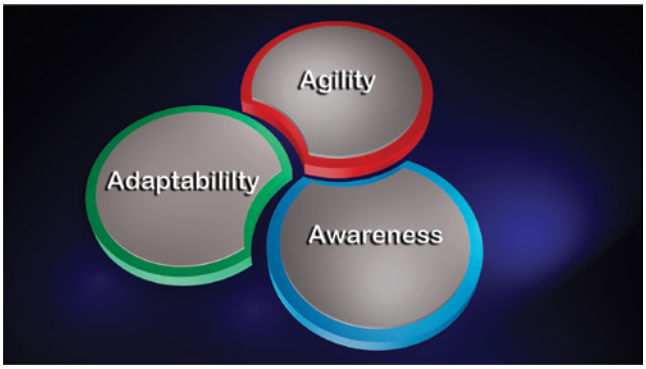


Figure I.6: Software Architecture Social Intelligence Pillars

Awareness is a unique social talent that can be leveraged to cope with complex social, business, and technological changes and challenges thrown your way. The term *adaptability* stands for versatility—an attribute that describes a skillful person with multiple talents for tackling organizational problems. Moreover, *agility* is a personal quality of a software architect who knows how to negotiate and compromise on technological solutions, resolve social conflicts, and collaborate with others in good faith.

Follow a Simple Process to Leverage Your Software Architecture Social Intelligence Skills

Your social intelligence skills can be instrumental in establishing working-related partnerships and alliances. To build a coalition of supporters and collaborators, consider these simple roadmap milestones: *search*, *connect*, *integrate*, and *cooperate*.

At the onset of this exertion, begin searching for candidates who understand your language and objectives and are willing to work together to achieve software architecture goals. While these individuals are typically found in close vicinities, such as in your organization, others can be spotted on social media and at technological conferences.

Once potential social partners are found, devote your time to connecting, raising their interest, and spurring enthusiasm for contributing to the organization and industry. Then share your knowledge. Learn from others. Interface and cooperate on strategies. *And always remember: you're not alone!*

Further Reading

The topic of software architecture social intelligence skills is covered chiefly in these two chapters:

- Chapter 4, "Self-Assessment for Software Architects," includes queries to evaluate an individual's communication, collaboration, and partnership formation skills required to promote software architecture strategies and contribute to technological transformation and innovation.
- Chapter 9, "An Outline for Software Architecture Job Interview Questions," introduces potential interview behavioral questions, preparing candidates to demonstrate communication, interpersonal relationship, and leadership capabilities.

You're Career-Driven

It's critical to carve out a long-term plan, a strategy that reflects your talents and capabilities. Equally important, stay attuned to your individual preferences, such as the types of duties that you'd like to fulfill and contribute to a specific sector and industry. However, focusing merely on your career agenda or promoting individual interests would never contribute to solving organizational problems or boosting your software architecture performance.

Remember that your preliminary duty is to collaborate with co-workers and partners to support business objectives—a vision greater than your aspirations. In software architecture, there is nothing nobler than teaming up with stakeholders to promote organizational culture, influence the outcome of business transformation, and accelerate technological modernization.

Carve Out a Software Architecture Career Strategy

A software architecture career strategy is a long-term plan that spells out incremental steps to achieving professional milestones and goals. Each milestone

is an important landmark, a checkpoint for evaluating your professional progress and achievements. A career milestone can also mark a turning point, perhaps a change in direction or adjustment to your software architecture employment strategy.

The software architecture career strategy's goal should not be considered your last professional occupation. On the contrary, in a long-term career time span, there may be multiple goals to pursue. Again, each milestone assessment should determine the next career step to conquer.

Moreover, a software architecture career strategy ought to be realistic. And professional development in the field must be gradual and feasible. The journey to achieving career goals should be devoted to knowledge acquisition, self-improvement, and delivering best-of-class software products and the architecture of their hosting environments.

Knowledge acquisition refers to the incremental learning and practice of software architecture disciplines during career employment. Specifically, business and technical knowledge are acquired through years of hard work, research, and studies. *Self-improvement* is related to the knowledge acquisition process. But it is affiliated chiefly with self-motivation and the individual appetite for improving software architecture capabilities.

TIP **Bottom line: a software architecture career strategy encompasses a gradual and self-challenging agenda that should be reevaluated at every milestone.**

Software Architecture Career Strategy Perspectives

Throughout the years, the software architecture field has grown immensely in scope. Professionals choose to focus on different architectural practices and disciplines. Some individuals pursue the leadership and governance route, while others focus merely on the technological aspects of software architecture roles. As illustrated in Figure I.7, this book centers on four chief software architecture career planning perspectives: social, technology, strategy, and leadership.

Social-Driven Career Perspective Consider this employment avenue if you seek to promote your professional objectives by forming productive alliances with collaborating partners and executives to provide business and technological solutions.

Technology-Driven Career Perspective If you focus merely on your technical skills, pursue this career path by applying software architecture capabilities and experience to provide business and technological solutions.

Leadership-Driven Career Perspective Choose this career path if you possess management skills and seek to focus on promoting enterprise culture,

steering technological transformation, and establishing institutional best practices, standards, and policies.

Strategy-Driven Career Perspective This role is for you if you look to influence enterprise business and technological evolution, foster digital transformation, develop organization-wide road maps, and align business strategies with software architecture strategies.

Figure I.7: Software Architecture Career Strategy Perspectives

Further Reading

A well-planned career path is a roadmap for a successful software architecture journey. But a career strategy is not the only ingredient for a flourishing occupation. Knowledge acquisition and carving out a winning strategy for software architecture interviews can indeed yield a lifetime of prosperous employment.

- Chapter 3, "Career Planning for Software Architects: A Winning Strategy," depicts the four career-driven perspectives that can impact organizational culture: social, technology, management, and strategy.
- Chapter 8, "Preparing for a Software Architecture Interview: A Winning Strategy," introduces a job interview preparation model that includes two different strategies to consider: interview defense and attack plans.
- Chapter 9, "An Outline for Software Architecture Job Interview Questions," presents potential software architecture questions that can increase the odds of acing an interview. They are grouped into ten different categories, such as technical, behavioral, social, problem-solving and decision-making, software architecture life cycle, and more.

You Trust Your Innate Talents

You undoubtedly bring a slew of talents instrumental in providing effective software architecture solutions to organizational problems. Moreover, you know

that these personal traits successfully contribute to your employment duties. You may have wondered if these individual aptitudes were with you at birth, or perhaps you've learned them on the job.

Numerous scientific studies submit that the talents you have been carrying since birth are recognized as innate traits—skills not necessarily learned through experience. These are affiliated with primal instincts—natural survival abilities such as endurance, social bonding, adaptability, enthusiasm, and more. We often employ them to endure economic hardships, social challenges, or natural calamities.

However, there is no indication that these survival abilities cannot be learned and honed during a lifetime, career journey, or professional training. And it has become evident that combining innate talents with on-the-job experiences improves the ability of software architects to deliver pragmatic and potent solutions. For example, software architecture capabilities to construct powerful applications and systems typically depend on professional traits such as balanced decision-making, effective problem-solving, and good taste.

Employ Innate Traits to Advance Business and Technological Missions

It is no secret that tempestuous organizational issues often challenge software architects. Some are affiliated with the struggle to advance software architecture roadmaps, visions, missions, and strategies. Fostering and maintaining technological leadership is another difficulty that software architects wrestle with. Facing stiff resistance to business change or technological modernization initiatives is another predicament that must be tackled.

Employ your communication, patience, and self-discipline capacities to alleviate unnecessary conflicts. Respect the diversity of ideas, concepts, and solutions your co-workers, managers, and partners propose. Consider their diverse approaches to solving software development and integration problems. Most importantly, stay tuned with the four innate talents that can enhance your decision-making capabilities (as illustrated in Figure I.8): *creativity*, *imagination*, *software design aesthetic*, and *curiosity*.

Avoid Self-Induced Software Architecture Blindness

Ignoring your innate skills when you need them the most promotes business stagnation, delays technological standardization, and stalls applications and systems modernization. There is nothing riskier to business development than the underutilization of fundamental innate skills, such as creativity and imagination.

Creativity and imagination are all about the enablement of business opportunities. They are the bedrock of every software architecture implementation

that allows the business to flourish and win the competition. On the other hand, curiosity is an essential innate gift that galvanizes research and studies and ultimately encourages perfection. Finally, the design aesthetic is an innate skill that entices consumers to buy goods, acquire services, and look forward to the next line of innovative products.

Figure I.8: Four Leading Innate Talents

TIP **Do not engage in self-induced software architecture blindness by overlooking your innate traits.**

Further Reading

Refer to Chapter 5, "Employing Innate Talents to Provide Potent Organizational Solutions," to learn more about the chief innate gifts that software architects should leverage to mitigate enterprise challenges and successfully promote software architecture agendas. This chapter also elaborates on a variety of methods to boost software architecture creativity, imagination, good design taste and aesthetics, and curiosity.

Part

1

Software Architect Capability Model

In This Part

Chapter 1: Software Architect Capability Model