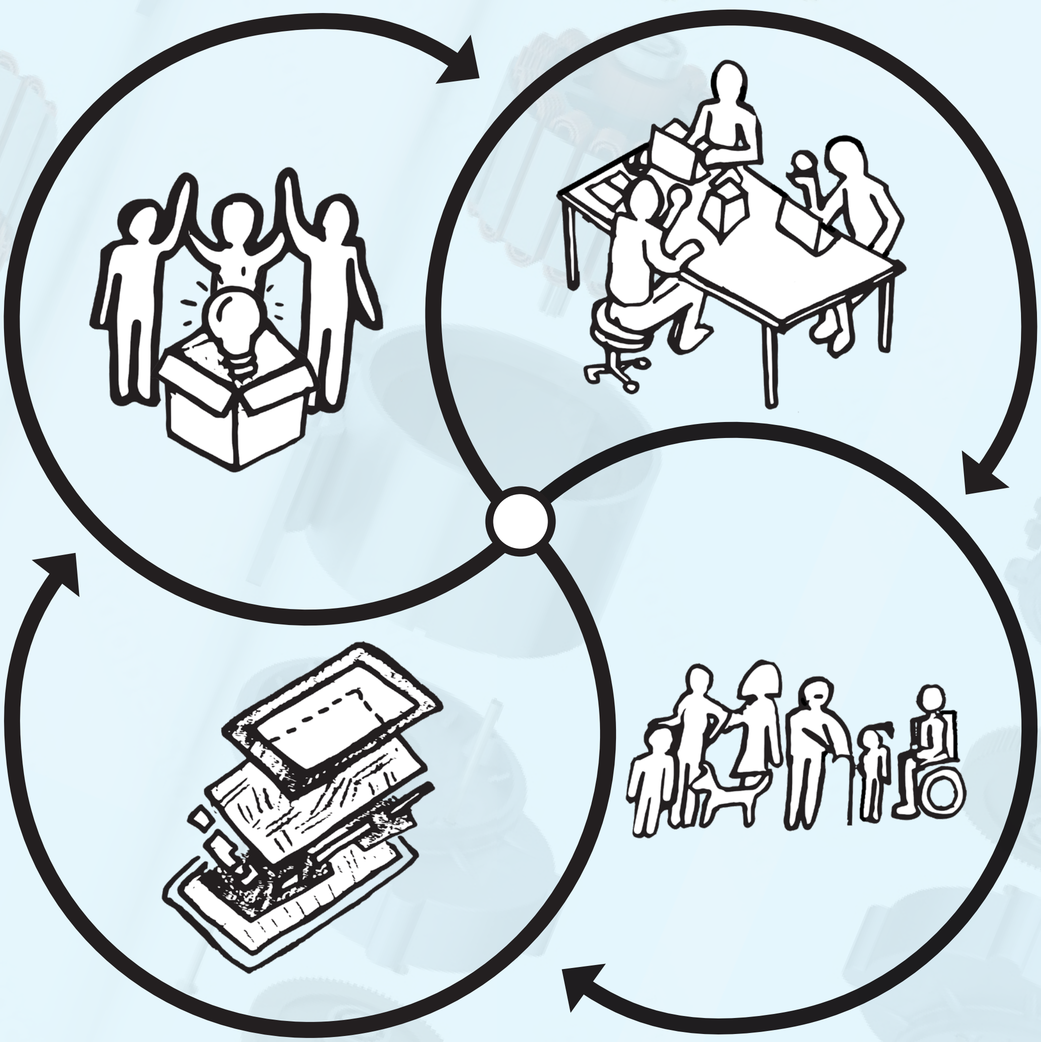
PRODUCT DESIGN BEST PRACTICES



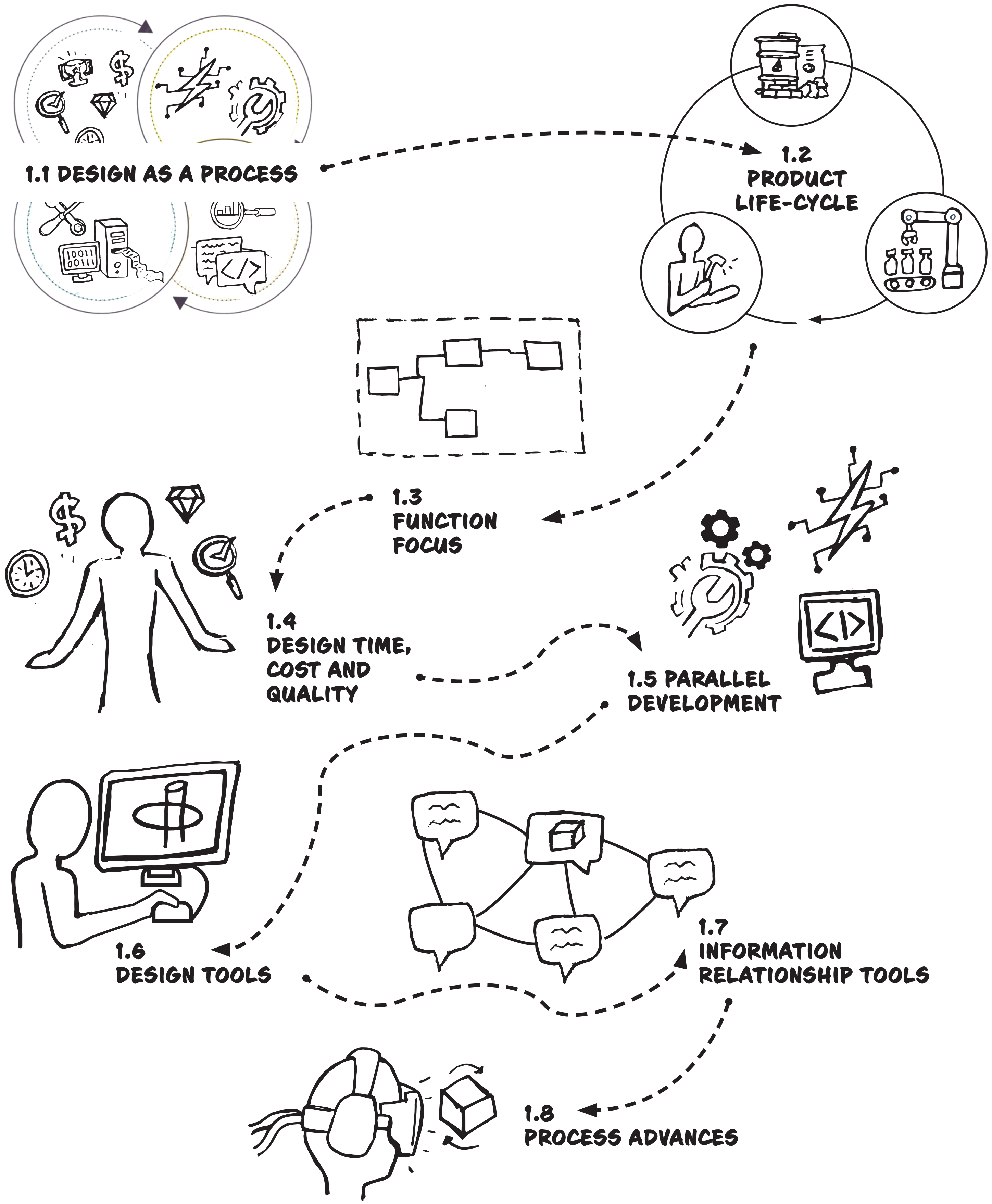
50 INDUSTRY-INSPIRED BEST PRACTICES FOR QUALITY PRODUCTS

David G. Ullman and Joshua D. Summers with contributions by Jeremy Fielding, Sr.

The 50 industry-inspired Best Practices are grouped into 10 Sections.

Click a section below to see the Best Practices (BP).

INTRODUCTION TO THE DESIGN PROCESS	**SECTION 1.** INTRODUCTION TO PRODUCT DESIGN BEST PRACTICES	**SECTION 2.** DESIGN PROCESS FUNDAMENTALS	**SECTION 3.** DESIGN TEAMS
BUILD UNDERSTANDING	**SECTION 4.** BUILDING PROBLEM UNDERSTANDING	**SECTION 5.** PLANNING AND MANAGING THE DESIGN PROCESS	
GENERATE CONCEPTS	**SECTION 6.** ALTERNATIVE GENERATION		
EVALUATE CONCEPTS	**SECTION 7.** EVALUATE FOR UNCERTAINTY	**SECTION 8.** EVALUATE FOR PERFORMANCE	**SECTION 9.** EVALUATE FOR X
POST DESIGN CONCERNS	**SECTION 10.** POST DESIGN CONSIDERATIONS		



BACK TO TABLE OF CONTENTS

SECTION 1.0

Introduction to Product Design Best Practices

The eight best practices in this section, shown as icons in Fig. 1, answer the following questions:

- What is the design process?
- Why is a systematic process useful?
- How does the design process affect the entire product's life cycle, from production to use to disposal or recycling?
- Why is the design process driven by the desired functionality?
- How can a design team balance product cost, development time, and quality?
- How can mechanical, electronic, software, and manufacturing systems be designed in parallel?
- What design and information relationship tools are available to support the design process?
- How will future developments change the design process?

"If you can't describe what you are doing as a process, you don't know what you're doing."

—W. EDWARDS DEMING

BEST PRACTICE 1.1

Successful Product Development Organizations

Have a Good Appreciation for the Design Process.

BEST PRACTICE KEY CONCEPTS

- The design process is the managed structure of people's knowledge so they can make the best possible design decisions, successfully fulfilling a need with objects.
- Many different design processes are used.
- Many factors make product design difficult.
- The design process is highly recursive; there are usually tasks within tasks.
- The design process is iterative, even in the most mature situations.

The Goal of This Best Practice

This best practice describes how organizations view the design process, introduces some typically used terminology, and develops what makes design so challenging. The quality of the design process used in an organization determines the length of product development, the cost of development, the cost of the product itself, and the quality of the product. Successful organizations are very process-aware, working to understand how effectively they design products and what needs improvement.

What is a Design Process?

Design activities result in hardware, electronics, and/or software to fill some needs. Whether the final object is a bookshelf or a space station, it results from a process that combines people with their knowledge, tools, and skills to develop a new creation. This process requires time and money, and if the people are good at what they do and their work environment is well structured, they can do it efficiently. Further, if they are skilled and make good design decisions, the final product will be well-liked by those who use and work with it—the customers will see it as a quality product.

The **design process** is the managed structure of people's knowledge so they can make the best possible design decisions, fulfilling a need with an object (as illustrated graphically in Fig. 1). The design process begins with the need for an object, a system, or some code on a chip within a system. This "need" is the difference between the current state of something and its desired state. People bring their knowledge and experience to develop concepts addressing

BEST PRACTICE 1.2

Successful Product Development Organizations

Show Concern For The Entire Product Life Cycle.

BEST PRACTICE KEY CONCEPTS

- The design process not only gives birth to a product but is responsible for its life and death.
- Decisions made by designers affect virtually all the stages in a product's life cycle.
- There are four stages in a product's life cycle: Design, Production and Delivery, Use, and End of Life. The designer must address all four.
- The Design stage is further broken into phases: Product Definition, Project Planning, Conceptual Design, and Product Development.
- The Production and Delivery stage consists of Manufacture, Assemble, Distribute, and Install.
- A product is often designed for many different uses.
- The end of life is usually a combination of the product being retired (taken out of use), then some combination of disassembly, disposal, recycling, or reuse.
- Modern products are designed cradle-to-cradle.

The Goal of This Best Practice

Successful organizations consider the design process in terms of the product's life cycle. Every product's life evolves through four stages, shown in Fig. 1.

The first stage concerns the product's development, which is the focus of these best practices. The second stage is the production and delivery of the product to the customer. The third is the product's use by the customer. The final stage focuses on what happens to the product after it is no longer useful. The first stage is the domain of the designer. But how the product fares in all the other stages is a direct consequence of decisions made in this first stage. So, in effect, the design process not only gives birth to a product but is also responsible for its life and death.

Design problems have many satisfactory solutions but no clear best solution.

BACK TO TABLE OF CONTENTS

BEST PRACTICE **1.3**

Successful Product Development Organizations

Focus on Function During Product Development.

BEST PRACTICE KEY CONCEPTS

- Function is what an object does.
- Action verbs communicate function.
- During design, function is in terms of the product's desired performance.
- As form evolves, function is in terms of the product's behavior, what it actually does. Behavior is realized performance.
- Mechanical design engineers work from function to form, where the form can be of many different shapes and alternative materials.
- Electronics engineers work on functions in various fixed forms (PC boards, components, and connecting wires).
- Software engineers operate exclusively on function.

The goal of this best practice

Successful organizations develop their products from function to form. In this Best Practice, the basis for this evolution is developed.

What is "Function"?

Successful organizations focus on function before they detail the form. **Form** is what an object is, including shape and materials; its **function** is what it does. Many objects are named for their function: screwdrivers, lawnmowers, hairdryers, resistors, and code functions (self-contained modules that accomplish a specific task). Function is communicated by action verbs (underlined). Screwdrivers allow the user to drive screws into another object. Resistors resist the flow of electricity. The code function "tan(X)," for example, is to calculate the tangent of the argument "X."

Software is totally functional; it has no physical form. Electronics generally provide function through a limited array of forms (resistors, capacitors, PC boards, and the wires connecting them). The interplay of form and function becomes interesting and challenging with physical objects, where the form that provides a function can be of many different shapes and materials.

Consider the handlebar of a bicycle, such as the one shown Fig. 1. The handlebar is a bent piece of tubing, a single component that serves many functions. It enables the rider to "steer the bicycle" ("steer" is an action

BACK TO TABLE OF CONTENTS

BEST PRACTICE 1.4

Successful Product Development Organizations

Consciously and Continuously Balance Product Cost, Development Time, and Quality.

BEST PRACTICE KEY CONCEPTS

- Engineers cost little but have a significant impact on product cost and quality.
- Design decisions affect the cost of a product as much as the manufacturing processes used.
- Product cost is committed early in the design process.
- Product quality is a combination of:
  - It works as it should (function).
  - It lasts a long time (reliability).
  - It is environmentally friendly (sustainability).
  - and other factors defined by the user.
- Best practices help keep time, cost, and quality in balance.

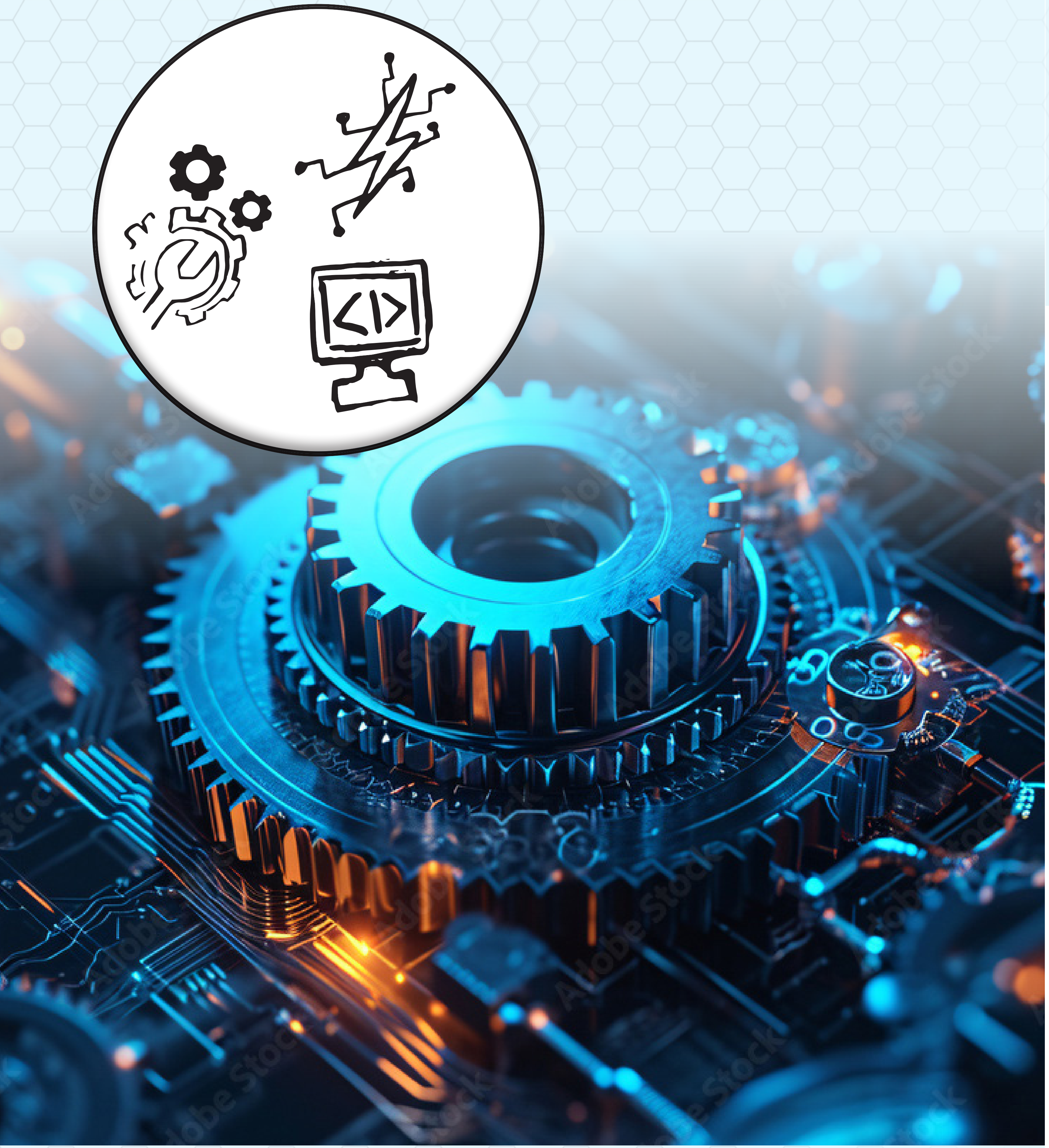
The Goal of This Best Practice

Successful product development organizations focus on three measures of design process effectiveness: the **cost** to develop and produce the product, the **quality** of the result, and the **time** to get it to market. Regardless of the product—whether it is a bookshelf for a dorm room, a complex luggage handling system at an airport, some small subpart of a larger product, or just a small change in an existing product—the customer and management always want it cheaper (lower cost), better (higher quality), and faster (less time). To a most control over these three factors, often depicted as a "Golden Triangle" (Fig. 1). Each organization has different priorities as to which of these are most important based on market segment, industry, competition, and other factors. However, all organizations consider tradeoffs between them, such as accepting a lower quality solution to make it to market faster or accepting higher costs to achieve better quality. These tradeoffs challenge engineers constantly.

Cost

The cost of designing a product is

BACK TO TABLE OF CONTENTS



BEST PRACTICE 1.5

Successful Product Development Organizations

Develop Mechanical, Electronic, Software, and Manufacturing Systems Concurrently.

BEST PRACTICE KEY CONCEPTS

- Most modern products rely on mechanical, electronic, and software to make them function.
- Mechatronic engineering is the simultaneous design of a product's mechanical, electronic, and software elements.
- No single person is an expert in all these fields; therefore, teams develop virtually all products.

The Goal of This Best Practice

Most modern products rely on mechanical, electronic, and software objects to make them function. Innovative companies know how to manage the development of these technologies concurrently – simultaneously in a synergistic and supportive manner. This is often referred to as mechatronic or concurrent design.

The term "**concurrency**" can have multiple meanings in the design process. The first, used here, is the simultaneous consideration of mechanical, electrical, and software objects during the design process. This is also known as mechatronic design. A second meaning is for Integrated Product Development (IPD), or "simultaneous engineering," where the various business functions of sales and marketing, finance, support, and production are all considered simultaneously.

Finally, ESA, the European Space Agency, defines concurrent engineering as (bold added to stress concepts in the best practices): *Concurrent Engineering (CE) is a* ***systematic approach*** *to integrated product development that emphasizes the response to* ***customer expectations****. It embodies* ***team values of cooperation****, trust, and sharing in such a manner that* ***decision making*** *is by consensus, involving* ***all perspectives in parallel****, from the beginning of the* ***product life-cycle****."*

To explore concurrent systems, the following section discusses many options for providing function. This is followed by examples to help make the point.

BACK TO TABLE OF CONTENTS

Successful Product Development Organizations

Use Modern Design Tools to Support the Product and the Process.

BEST PRACTICE KEY CONCEPTS

- There are many types of tools that support product design:
    - Communication Tools
    - Planning Tools (Gantt charts, Road Maps)
    - Lifecycle Support Tools (Product Lifecycle Management)
    - Form Management Tools (CAD, Solid Modelers)
    - Performance Analysis Tools (FEM, CFM, Kinematics)
    - Synthetic Design Tools (Genetic Algorithms)
    - Human Interface Simulation Tools (Virtual/Augmented/Mixed Reality, Holographs, Haptics)
    - Augmented and Virtual Reality
    - Information Relationship Tools (Mind maps, DSMs, flow charts)
    - Measurement Tools (thermal imaging, laser scanning)
- Tools take learning effort to be useful.

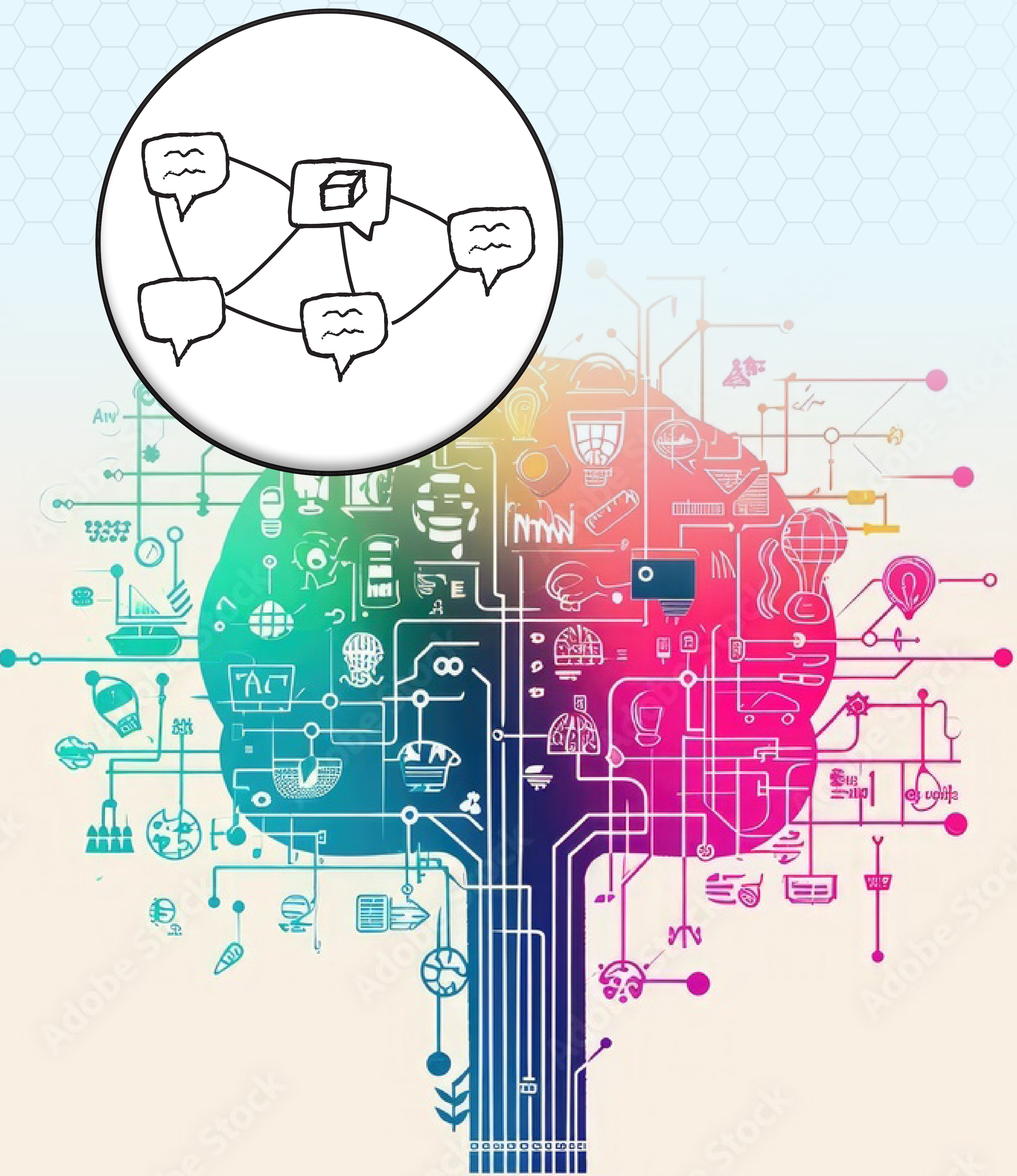
The Goals of This Best Practice

While this book is a compendium of methods used by successful organizations to support the design process, many of these methods are made easier or enabled using design tools. A **design tool** is a software or physical object that helps execute a design method or adds significant information during the design process. The right physical and computer-based tools help teams do their jobs effectively.

Computer-based tools play an integral role in the design process. Before the computer revolution of the 1980s, the only "tools" were pencil and paper, slide rules, physical models, and physical test equipment. Then VisiCalc, the first commercial spreadsheet, was introduced in 1981, and AutoCAD, the first widely used CAD system, became available in 1982.

Analysis tools like Finite Element Analysis (FEA) and Computational Fluid Dynamics

BACK TO TABLE OF CONTENTS



BEST PRACTICE 1.7

Successful Product Development Organizations

Use Information Relationship Management Tools.

BEST PRACTICE KEY CONCEPTS

- Mind Maps help develop relationships amongst people, objects, places, concepts, or events.
- Affinity Diagrams are ideal for generating, organizing, and consolidating product or process information.
- Flow Charts help organize anything that flows, such as information, energy, control, materials, task work, and data.
- DSMs are used to determine the sequence of tasks or functions and to cluster items into work or structure modules.

The Goal of This Best Practice

Product design is the evolution of information punctuated by decisions. The "information" begins with a description of the product need and ends with a complete picture of the product in terms of drawings, models, analyses, and supporting documentation. For many products, managing the evolution of information and the relationships it involves is as important as managing the representation of the products themselves.

Two types of information are of key concern: the design process and product function. **Design process relationships** form the structure of the best practices in this book. It is the management of the need as it is evolved into design requirements and specifications, which spawn concepts that are matured into final products.

Product function is the transformation and flow of materials, information, control, and energy. Designing these transformations is designing in an information-relationship space.

The four methods in this Best Practice can be applied in many ways. Much of what is described here can be done with paper and pencil, but often, computer tools make developing information and relationships easier.

Mind Maps

Mind maps help develop relationships among people, objects, places, concepts, or events. The relationships can be functional (the flow of information, energy, control, or materials), form, influence, content, and ideas. Mind maps can be used to support many of the best practices but are especially

BACK TO TABLE OF CONTENTS

BEST PRACTICE 1.8

Successful Product Development Organizations

Develop and Leverage Technological Advances.

BEST PRACTICE KEY CONCEPTS

- Artificial Intelligence, additive manufacturing, and internet communication are changing the design process in unforeseeable ways.
- Design process tools and methods constantly evolve to utilize these new technologies, generating new best practices.

The Goal of This Best Practice

The world is in the early stages of the 4th industrial revolution. The first revolution, known as the Industrial Revolution, began at the end of the 18th century with the use of steam and water for power and iron and steel in bridges and other equipment. Early in the 20th century, the second revolution leveraged mass production and electricity for energy. The third revolution began in the 1970s with integrated electronics, computers, and internet technology affecting products and processes. Each of these revolutions impacted the types of products and how they were designed.

With the introduction of Artificial Intelligence (AI), the Internet of Things (IoT), additive manufacturing, societal connectivity, generative design, and digital twins, the fourth industrial revolution—"Industry 4.0"—allows the creation of smart products and processes. It will profoundly affect 1) the products that are designed, 2) the processes to manufacture them, 3) the tools and methods used to design them, and 4) the design process itself.

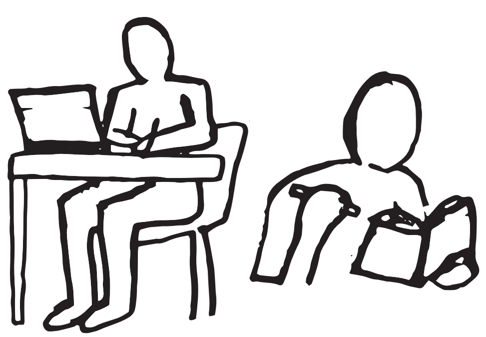
In short, Industry 4.0 = integrated things, integrated information, integrated decision-making, integrated people, and integrated services.

Successful organizations will respond to and develop tools and methods to use new technologies in unforeseeable ways. This Best Practice addresses some evident Industry 4.0 technologies that will impact the design process and the resulting products. The topics covered are represented in Fig. 1, where the arrows indicate the foreseen influence on each other and how they may affect the design process.

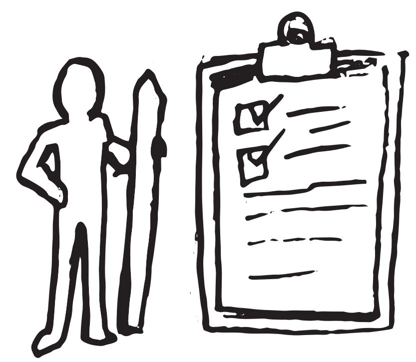
Artificial intelligence

Many evolving technologies rely on AI systems which can recognize a pattern and integrate the implications of the pattern into the existing knowledge, much as people do. There are four core components in this definition (see Fig 1).

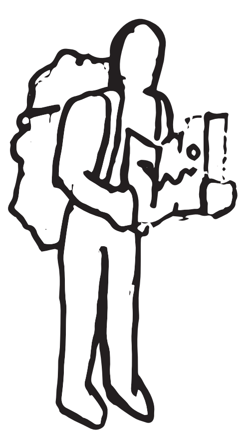
BACK TO TABLE OF CONTENTS



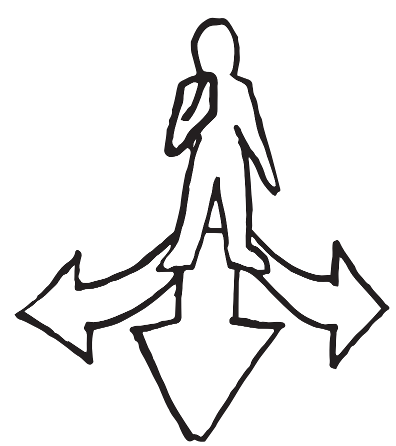
2.1 DESIGN LEARNING



2.2. DESIGN DOCUMENTATION



2.3 UNCERTAINTY AND RISK



2.4 DECISION MAKING

BACK TO TABLE OF CONTENTS

SECTION 2.0

Design Process Fundamentals

In this section, the fundamental activities supporting the design process are addressed: learning, communicating managing uncertainty and risk and making decisions, as shown in Fig. 1. Best practices covering these four are developed, forming the core of the rest of this book. These may seem like something that can be skipped, but they are the pillars on which a successful design process is built, with each having unique design process elements in them.

The best practices in this section answer the following questions:

- Why is designing a learning process?
- Why are communication and documentation so important during design?
- How can we manage uncertainty and risk?
- How can we make rational and transparent design decisions?

BEST PRACTICE **2.1**

Successful Product Development Organizations

Treat Design as a Learning Process.

BEST PRACTICE KEY CONCEPTS

- Individual engineers, the team, and the organization learn throughout the design process.
- Learning is a spiral, building on what was learned before.
- Every action taken during the design process results in new knowledge about the product and process.
- What is learned is reflected in the fidelity of the analytical and physical models.

The Goal of This Best Practice

Engineers in successful organizations embrace design as a continuous learning activity. While defining the problem, they learn about the stakeholders and their needs and desires. While developing concepts, designers learn about the created objects through testing, simulation, and prototyping. Along the way, they also learn new tools and methods to help execute projects quickly and effectively.

Successful organizations are constantly learning about new technologies that may be useful. Both during and at the project's end, engineers reflect on what went well, how different tools supported the process, and how the teams communicated and worked together.

Fig. 1 illustrates where learning happens throughout the entire project. Several learning opportunities are illustrated in the figure. During the process of building problem understanding, the evolving "key problem" is learned and articulated. This understanding is based on gaining knowledge about the stakeholders, their needs and desires, and past projects, as well as benchmarking existing similar products. The understanding is "evolving" because learning about the problem occurs as a spiral throughout the design effort.

As problem requirements are identified, the underlying design objective is uncovered. At first, the project may appear to be a new automation system for manufacturing, but the key problem might be a more robust approach that is insensitive to changing assembly operators. This might be an automated system, or it could be a simplified, decoupled assembly process, or it might be a fixture. The key problem is often not the same as the original design problem.

During the heart of the design process—alternative generation and evaluation—

BEST PRACTICE 2.2

Successful Product Development Organizations

Value Documented Design Information.

BEST PRACTICE KEY CONCEPTS

- Design Information ranges from sketches on paper to the hardware and software of the final product.
- Information not documented is often lost to the individual and always lost to the organization.
- Document all artifacts and decisions, as these are part of the organization's intellectual property.
- Documents need to capture what is in the engineers' heads and communicate it to other engineers.

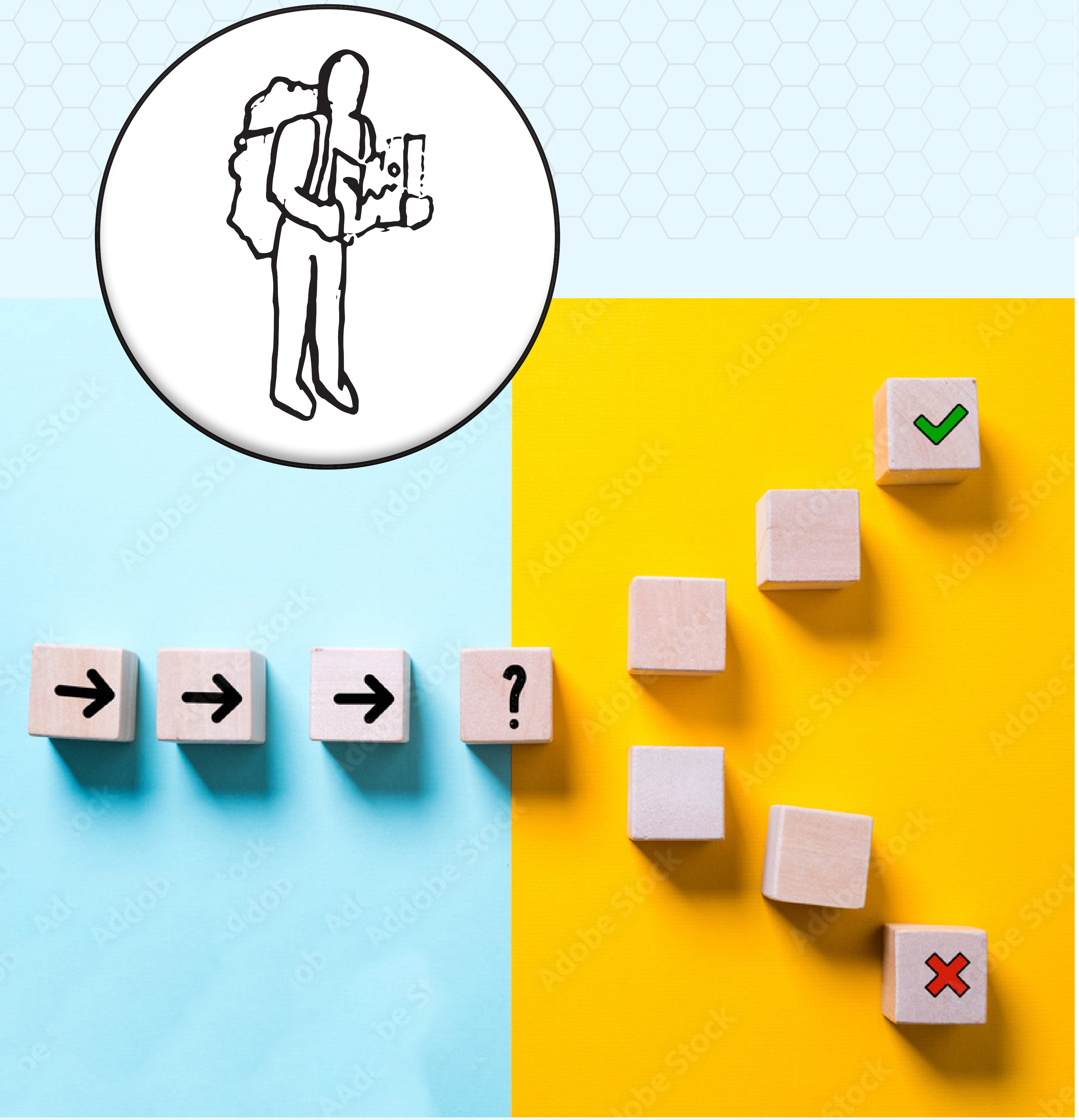
The Goal of This Best Practice

If information is not captured, it is lost. During the design process, engineers capture information using a variety of documents. This Best Practice introduces the elements of design information and the many ways to capture it. It would be helpful if there were a replay button so an engineer or organization could go back to see why a decision was made, what other options were considered, or who participated in designing a specific feature. Unfortunately, there is no such button. The reality is that the only information captured is documented information.

Successful organizations have robust means of capturing information and the decisions made throughout the design process. This documentation helps with communication within the team, creates a design history to use in future projects, enables individuals and the organization to continue to learn and build understanding, and records the organization's intellectual property.

All the information generated within the organization is **intellectual property**, defined as all the knowledge, drawings, documentation, test reports and other material generated during the design process that have value to the organization. Engineers have an ethical responsibility to both capture and protect this information during and after the development process.

Fig. 1 shows the "documentation onion." At the center are the elements of communication required for every document. The second ring contains the typical documents produced during the design process, such as models, worksheets, and reports. Each document should contain the elements to communicate completely. Finally, the outer ring contains the design



Successful Product Development Organizations

Manage Information Uncertainty and Risk.

BEST PRACTICE KEY CONCEPTS

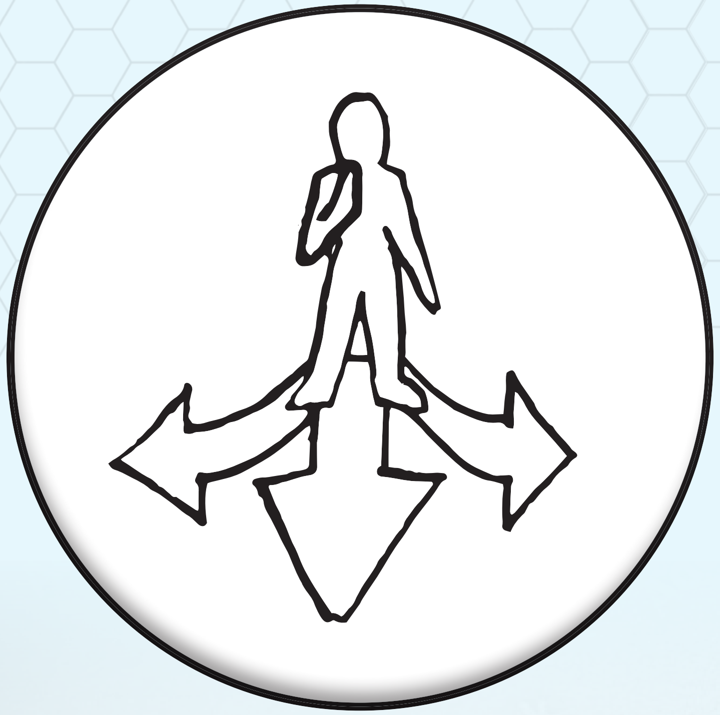
- VUCA is widely used to describe information uncertainty: VUCA = Volatile + Uncertain + Complex + Ambiguous.
  - ◊ Volatile information is changing and evolving.
  - ◊ Uncertain information has a distribution about a mean.
  - ◊ Complex information causes change in unexpected places.
  - ◊ Ambiguous information is subject to each individual's interpretation.
- VUCA leaves out two additional forms of uncertainty: incomplete information and Black Swans.
  - ◊ Incomplete information is missing key data.
  - ◊ Black Swans are unforeseen events that have a significant effect on the product.
- The combination of all forms of uncertainty is referred to as VUCA+.
- Product and project risk is dependent on managing VUCA+.

The Goal of This Best Practice

Mature organizations see product design as the evolution of uncertain information from need to product. How well the organization identifies and manages the uncertainty determines the final product's quality and the efficiency of the design process. In this Best Practice the types of uncertainty and associated jargon are explained. These concepts permeate all the best practices, so the discussion here will link to many others.

As technology and markets change, product development increasingly operates in an uncertain environment. For many years, the term "uncertainty" has meant different things in different disciplines. Recently, the Agile community has coined a term for elements of uncertainty: VUCA, an acronym for Volatile, Uncertain, Complex, and Ambiguous, as seen in Fig. 1.

While "VUCA" helps solidify what contributes to uncertainty, it has two problems. First, it has the term "Uncertain" in it, resulting in a circular definition. There is simply a lack of words in English to describe the different forms of uncertainty without using "uncertainty" (as defined below) as one of them.



BEST PRACTICE 2.4

Successful Product Development Organizations

Make Rational and Transparent Design Decisions.

BEST PRACTICE KEY CONCEPTS

- Design is a series of interdependent decisions.
- The activity of decision-making creates information and facilitates conversation and understanding.
- ALL decisions have five essential parts: understand the issue, develop measures, generate alternatives, evaluate alternatives, and decide what to do next.
- Addressing the basic decision-building blocks fosters learning.
- ALL decisions are based on VUCA+ information.
- Methods to help engineers make decisions include pro-con lists, pairwise comparisons, and decision matrices.
- Successful organizations know when decisions are final.

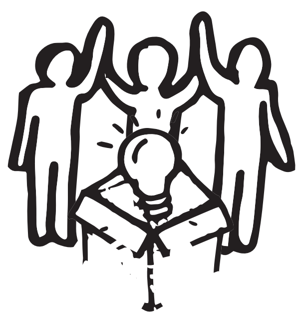
The Goal of This Best Practice

This Best Practice describes methods used by successful organizations to manage decisions. Before describing decision support methods, it must be realized that virtually all information is VUCA+, which makes decision-making difficult. Information remains Volatile (changing) as more is learned. It is Uncertain until drawings and code are final. Information is Complex because decisions about one part of a system may have unanticipated consequences for others. Information is Ambiguous because one team member's reality differs from another's. Additionally, information is incomplete because not everything is known, even when products are ready to be sold. Finally, Black Swans may be lurking, ready to add something unanticipated.

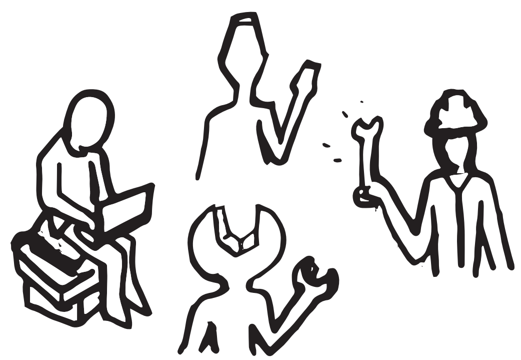
Regardless of the VUCA+ state, the need for action is a design reality. Time is passing, others need commitments, a manager demands an answer, or the competition is moving ahead, forcing the organization to make decisions based on VUCA+ information. There is never enough time and money to eliminate all uncertainty, and conflicting information will always exist.

Our goal here is to develop methods to manage these realities. Successful organizations recognize the tension around decision-making and realize that using the methods presented encourages communication, information development, and learning. They know that working through these methods is as valuable as the result.

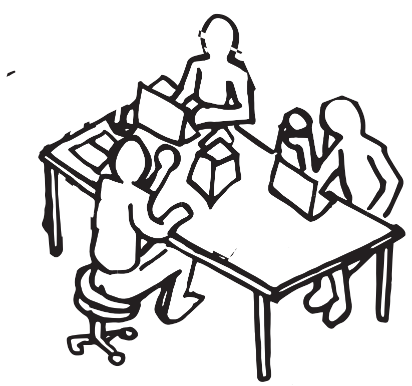
BACK TO TABLE OF CONTENTS



3.1 PRODUCT CENTERED TEAMS



3.2 ROLES, EXPERTISE, CREATIVITY AND STYLE



3.3 DESIGN ENVIRONMENT

3.4 TEAM HEALTH

BACK TO TABLE OF CONTENTS

SECTION 3.0

Develop Design Teams

While team building and team maintenance are not generally considered engineering activities, the quality of teamwork greatly affects the quality of the product, its cost, and time to market. Organizations that pay little attention to teamwork suffer, as do their employees.

The material in this section is broken into four best practices, as shown in the Figure to the left. More than most of the other sections, the best practices are interdependent, answering the following questions about teamwork:

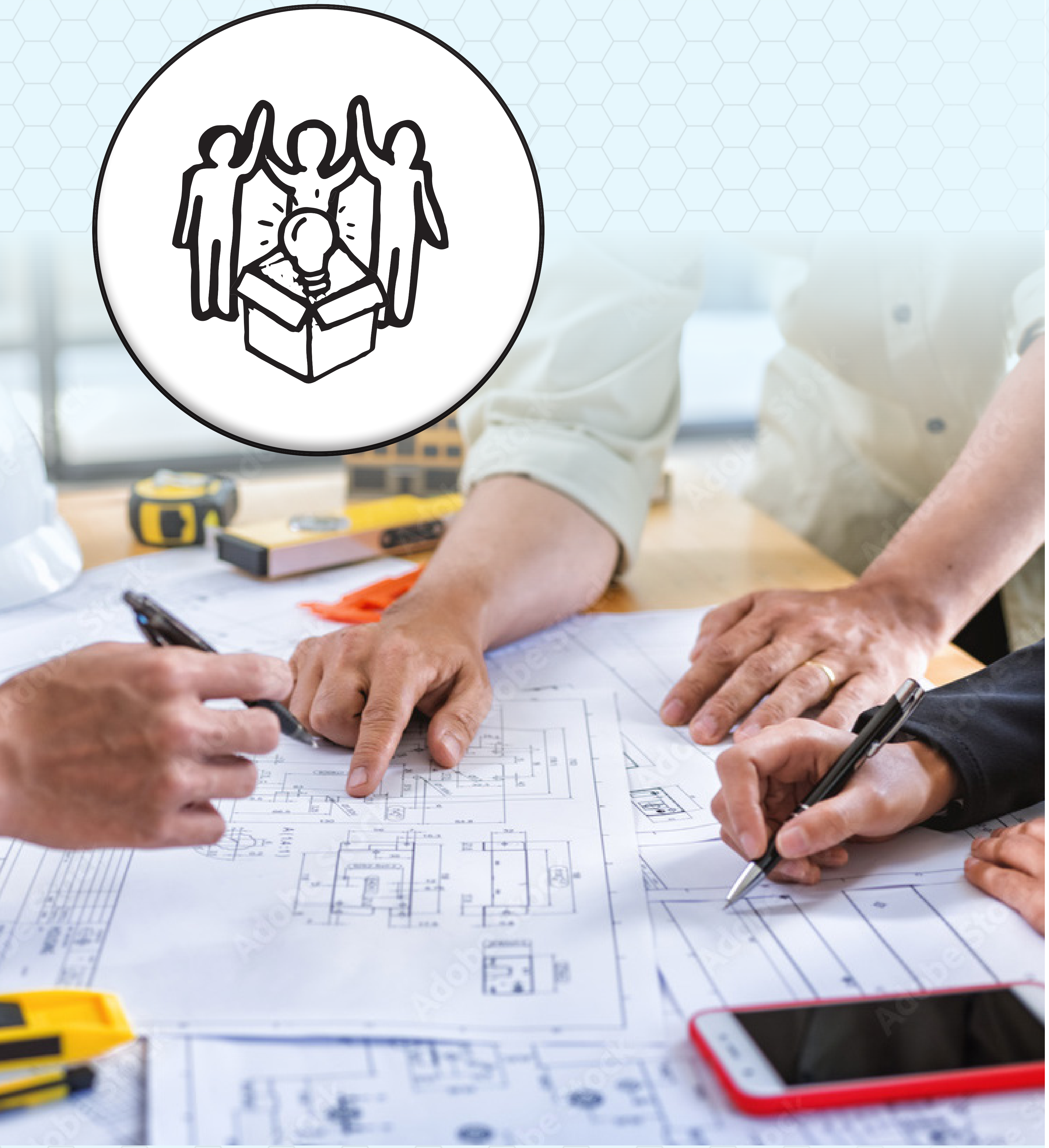
- Why use product-centered teams?
- How can the team support everybody's role, expertise, creativity, and decision-making style?
- What is a strong working environment?
- How can a team maintain good team health so everybody on it can do their best work and feel good about their contribution?

The first best practice focuses on what a team is and why teams are necessary for most product design work. Actually, it goes further by exploring team organization and goals.

The second is directed at who is on the team, their roles, the expertise and creativity they bring, and how the individuals solve problems - getting the best out of each other so that the team is more than the sum of its parts.

The third concerns the team's work environment, including the physical space they work in, the tools they use, and the work culture. This best practice focuses on where the engineering takes place physically and virtually.

The fourth best practice is how to maintain team health. While this topic is often overlooked in engineering education and professional training, it is vital to productivity and team members' feelings toward the team and the organization.



BEST PRACTICE **3.1**

Successful Product Development Organizations

Use Product-Centered Design Teams.

BEST PRACTICE KEY CONCEPTS

- A team is a group in search of a common understanding.
- Successful teams have identifiable characteristics.
- Simple team-building activities can set the pattern for team success.
- Often teams are part of a hierarchy of teams.
- It is helpful to have a team contract.

The Goal of This Best Practice

Successful organizations are composed of successful teams. A product development team is ideally 4-9 people with complementary skills committed to a common purpose, performance goals, and the approach for which they hold themselves mutually accountable. In contrast, the group members interact primarily to share information and make decisions to help everyone perform within their area of responsibility. However, an effective team is more than the sum of its parts. Important points about teams are:

- **Teams work collaboratively to develop new information**: Teamwork is central to success in engineering. Most design problems have many interdependent subparts that teams must solve collaboratively and concurrently. Teams bring together complementary skills and experiences needed to solve most engineering problems. Collaboration means more than just working together: it means getting the most out of the other team members.
- **Successful teams make decisions as a unit:** Teams develop decisions by collecting and considering all the relevant information leading to more robust decisions rather than decisions by authority.
- **Strong teams communicate to get the best from each individual**: Teams establish communication to support real-time information development and decision-making (i.e., problem-solving). Further, members need to ensure that others have the same understanding of design ideas and evaluations. It is challenging for people with different areas of expertise to develop a shared vision of the problem and its potential solutions, which requires good communication and developing a rich understanding of the problem.

BEST PRACTICE 3.2

Successful Product Development Organizations

Support Each Individual's Roles, Expertise, Creativity, and Style.

BEST PRACTICE KEY CONCEPTS

- There are many **roles** on a design team, and successful organizations ensure that all the important ones are filled.
- Engineers are on the team primarily for their **expertise**. Successful organizations ensure that the right expertise is on the team.
- Successful organizations foster **creativity**.
- Each member of the team has their own **problem-solving style**. Successful teams get the best out of all their members regardless of style.

The Goal of This Best Practice

Each team member fills a role, bringing their own expertise and creativity, and has a personal problem-solving style. For the team to reach its potential, each member must give their utmost while helping others achieve their best. Successful organizations create an environment that makes roles clear, formulates the best use of expertise fosters its growth, encourages creativity, and leverages each person's approach to solving problems.

Design Teams: Roles

Successful organizations build their design teams with members who fill many different roles. The roles vary with the product development phase and from product to product and company to company, and the titles differ as well. Each position on a team is described as if filled by one person. Many people may fill that role in a large design project, whereas, in a small project, one individual may fill many roles. In very small organizations, all the roles are often filled by two or three people. A typical team could look like the one shown in Fig. 1.

Product design engineer. The product design engineer fulfills the primary design responsibility (hereafter referred to as the design engineer). This individual must be sure that the team clearly understands the product's needs. The design engineer also ensures that the product develops and

BACK TO TABLE OF CONTENTS

BEST PRACTICE 3.3

Successful Product Development Organizations

Ensure the teams have a work environment that fosters success.

BEST PRACTICE KEY CONCEPTS

- A good physical workspace can help a team be successful.
- Engineers need a realistic workload without many distractions.
- Successful organizations enfranchise their teams to make decisions.
- The right tools help engineers to do their jobs effectively.
- People work best in a positive company culture where everyone wants to contribute and do well.

The Goal of This Best Practice

Work environments differ significantly among organizations. Successful product development organizations ensure that the physical environment is conducive to good design, and the virtual and organizational environments support the design teams. The environment set by the organization can be measured by how it manages the physical workspace, allocates workload, enfranchises teams, provides tools to get the job done, supports communication, and its overall attitude toward its employees.

The key elements of the team environment can be evaluated using a checklist, Fig. 1. Not all the items on the list apply to all teams, but addressing each item can measure the current quality of the team's working environment and help set goals for an improved environment.

Successful Product Development Organizations Provide a Physical Workspace Conducive to Success.

The structure of the physical workspaces for product designers is changing. For many years people worked in cube farms, as shown in the exaggerated cartoon in Fig. 2. Often, these were organized by discipline with the electrical engineers in one section (or even a separate building, plant, or country) and the structures people in another, software in a third, and so on. So, a team working on a product may be distributed across the room, building, or even countries.

BACK TO TABLE OF CONTENTS

BEST PRACTICE 3.4

Successful Product Development Organizations

Maintain Good Team Health.

BEST PRACTICE KEY CONCEPTS

- Team health means getting the best from the team and the individuals.
- Recognize the symptoms of poor team health and poor team member contribution.
- Base remedies on a clear identification of the causes
- Use retrospection as a part of continuous design team improvement.

The Goal of This Best Practice

Successful organizations do everything they can to get the best out of their engineers, which leads to better products, greater employee retention, and generally happier, more content workers. Poor team or individual performance can increase product cost, lower quality, or cause a product to be late to market. In successful organizations, issues are discovered before leading to these problems. Further, they are discovered and remedied within the team without management intervention.

Successful organizations prepare team members to recognize the symptoms, identify the causes, and apply remedies whether the cause is an individual or a function of team behavior.

To aid in this, many organizations have built-in meetings where these issues are addressed. Often called “retrospective” or “debrief” meetings, they are a look back at what went well and what did not in the team’s execution of the design process. These are effectively “design reviews” for the design process rather than the product.

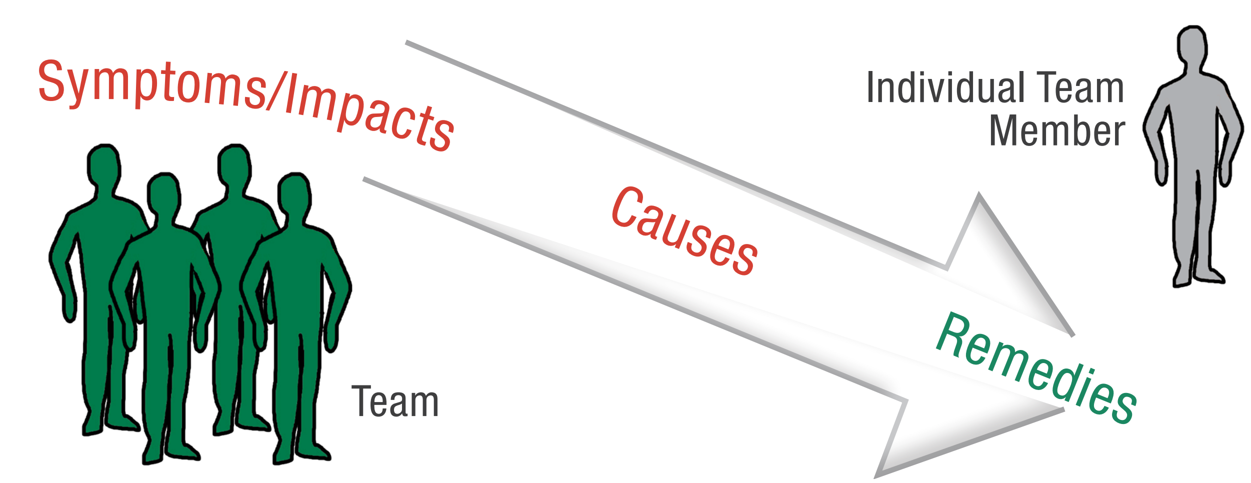
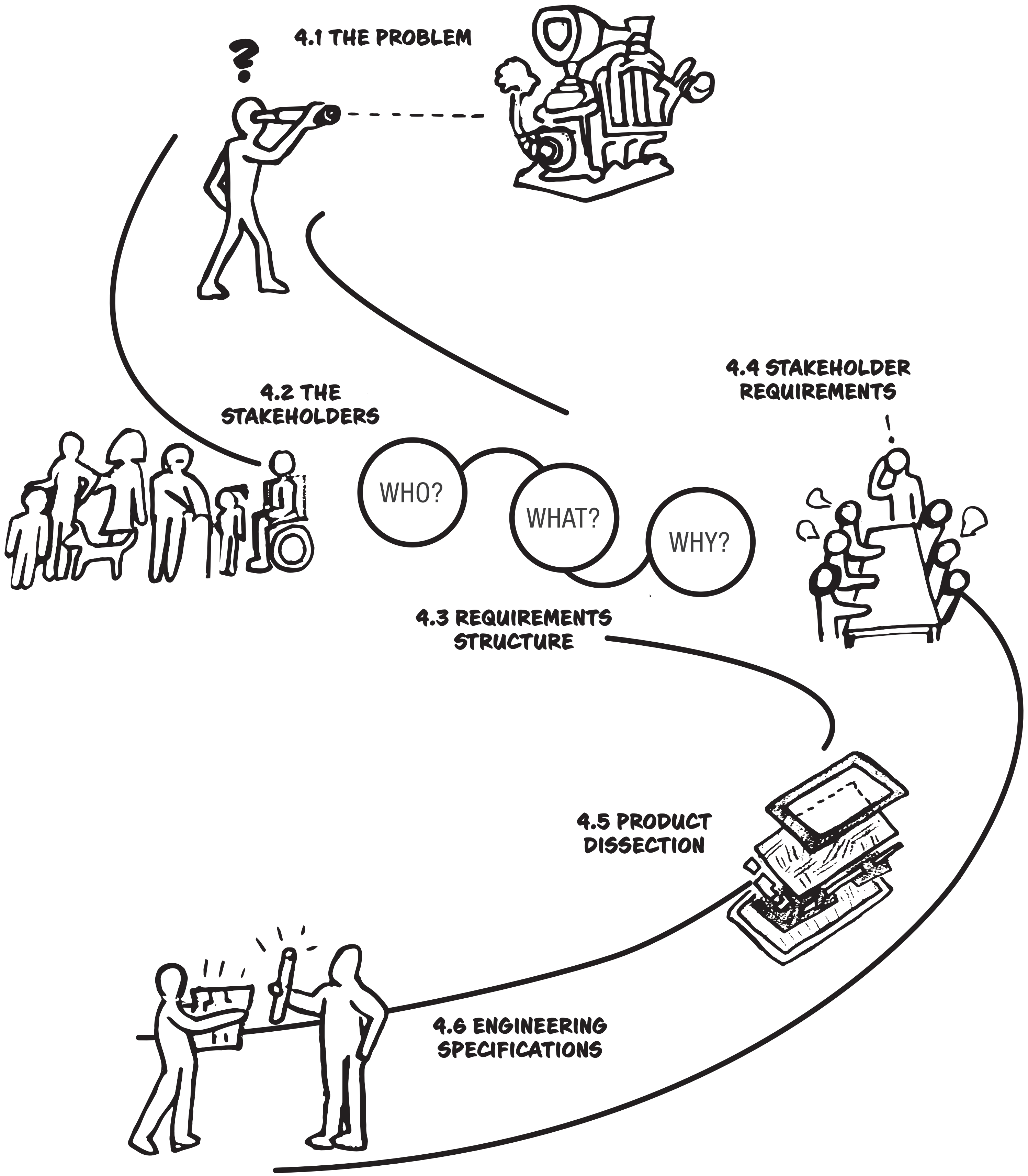


Figure 1. The flow of health symptoms to remedies.

BACK TO TABLE OF CONTENTS



SECTION 4.0

Building Problem Understanding

Understanding the problem is an essential foundation for designing a quality product. "Understanding the problem" means refining the mix of information that is initially known into concrete engineering specifications. When facing a new problem, some of the information will be vague "wants" or "desires," such as: "I want it to be fast and easy to control," and others will be specific such as, "It must be less than 80mm wide so it can fit in a pocket." Designing something "fast" and "easy" without knowing what these words mean is impossible.

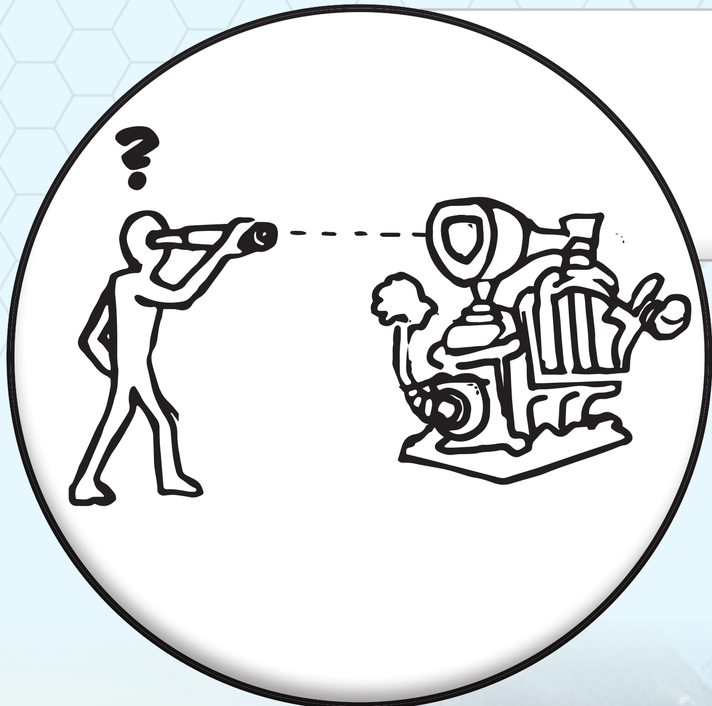
To achieve what is needed the following questions need to be answered no matter how large or small the problem:

- Does the team have a clear problem statement?
- Have the stakeholders been identified?
- What are the stakeholder requirements?
- What are the benchmark products and how do they meet the stakeholders' requirements?
- Are the engineering specifications sufficient to understand the problem?
- How much time should be spent on understanding the problem?

Best Practice 4.1 addresses understanding design problems. There are many different types, across many disciplines with varying levels of granularity and maturity. This is followed by attention to the stakeholders, the people and systems affected by the decisions made, in Best Practice 4.2. Then, what is wanted or needed must be captured and understood, covered in Best Practices 4.3 and 4.4. Also needed is knowledge about how the problem is currently being resolved and gaining this knowledge by dissecting existing products, Best Practice 4.5. Finally, the goal of understanding is developing engineering specifications, Best Practice 4.6.

Also covered in this Section is Best Practice 4.7, Quality Function Deployment, a widely used method that integrates the other best practices to build problem understanding. This is not shown in the figure, as it covers all the questions.

The bottom line for this Section is: if you don't know where you are going, you won't know when you get there.



"The problem is we don't understand the problem."

—PAUL MACCREADY

BEST PRACTICE **4.1**

Successful Product Development Organizations

Ensure Design Problem Understanding.

BEST PRACTICE KEY CONCEPTS

- There are many types of design problems.
- Design problems focus on all levels of product granularity, from entire products to minor details.
- Some problems depend on mature technologies, and others on those just evolving.
- The process of addressing problem understanding is the same regardless of discipline, type, granularity, or maturity.
- Work yet to be done on the design problem is called the Problem Backlog.

The Goal of This Best Practice

Design problems are usually categorized by their discipline (ME, EE, or CS), type (reuse, parametric, original, etc.), granularity (features to systems), and maturity. Regardless of how the problem is classified, the process of addressing it is the same.

Design problems are usually ill-defined and vague. Goals suggested by the stakeholders are often conflicting and inadequate. Successful organizations make sure they communally understand the problem and the goal. They establish a product backlog, an itemization of what is known about the problem and what needs to be learned, early. Further, they understand how much time they need to commit to refining their understanding using the other best practices.

PROBLEM CLASSIFICATION

Type	Granularity
Discipline	Maturity

Types of Design Problems

Most design situations are a mix of various problem types. For example, consider the design of a new consumer product that will accept a whole raw egg, break it, fry it, and deliver it on a plate. Since this is a new product, there will be much ***original*** design work to be done. As the design process proceeds, the parts will be ***configured*** relative to each other. To determine the thickness of the frying surface, the team will analyze the heat conduction of the frying component by ***parametrically*** changing the thickness, heat conduction, and other variables. And they will ***select*** a heating element and various fasteners to hold the components together from a catalog.

BACK TO TABLE OF CONTENTS

BEST PRACTICE 4.2

Successful Product Development Organizations

Keep the Stakeholders at the Forefront of the Project.

BEST PRACTICE KEY CONCEPTS

- A stakeholder is a person or organization that interacts (directly or indirectly) with the object being designed.
- Stakeholders are the main source of requirements.
- The primary stakeholders are customers and users.
- Two practical methods to find the stakeholders are Journey Maps and Stakeholder Checklists.
- Stakeholders may evolve and change their product needs.
- Providing safety for the stakeholders is a key design responsibility.

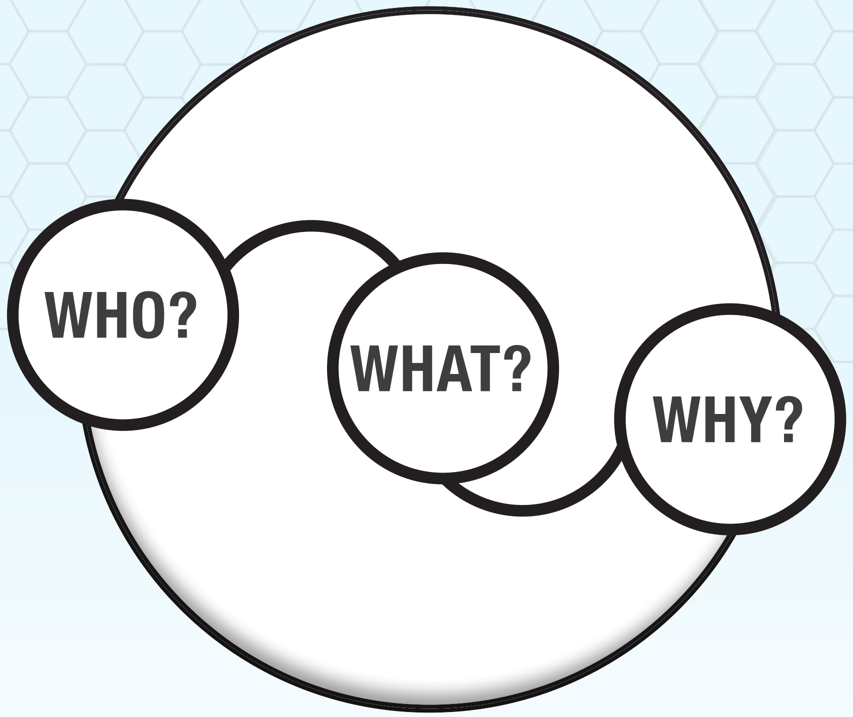
The Goal of This Best Practice

Problem understanding starts with stakeholder understanding. A **stakeholder** is any individual, group, or organization that interacts with the object being designed throughout its entire lifecycle. Customers, the people who buy the object, and users, those who use it, are often the **primary stakeholders**. The term "customer" is often used casually to mean both the purchaser and the user. But, sometimes, the purchaser of the product is not the same as its user. For example, an airline might buy airplanes from Boeing, but purchase thrust from GE and sell tickets to individual travelers. The people who buy tickets and the pilots are both users of the airplane, but not customers (they did not buy the airplane). Likewise, Boeing does not buy the engines from GE but designs their planes to accept these so that the airline can purchase the functionality of thrust from GE. GE still owns the engines and can repair, maintain, overhaul, or replace them.

For all design projects, whether for a complete product, a feature of one part, or a small segment of code, it is important to consider stakeholders both inside the organization (those who manage, manufacture, sell, distribute, and service the product) and external stakeholders, including purchasers and users. External stakeholders include the general public when considering the end of a product's life; a repairperson who must disassemble, diagnose, and repair the system; or the neighbor who can hear the air compressor running.

In fact, identifying the stakeholders is the first step in developing a good set of specifications and understanding the problem.

BACK TO TABLE OF CONTENTS



BEST PRACTICE 4.3

Successful Product Development Organizations

Work to Understand the Design Requirements.

BEST PRACTICE KEY CONCEPTS

- Design requirements are statements about what the stakeholders want.
- There are many types of design requirements.
- The stakeholders' interaction with the object directly determines its perceived quality and, to a great degree, its salability and success in the marketplace.

The Goal of This Best Practice

A major step in refining the problem backlog is finding the stakeholders' requirements - a description of what they want in the product. The importance of doing a thorough job here is made graphically clear in the Fig.1 cartoon. Everyone has a different view of what is needed, and it takes work to find consensus. Organizations that implement this Best Practice (and the others in this section) have matured from product-focused new product development to innovation centered on the customer experience.

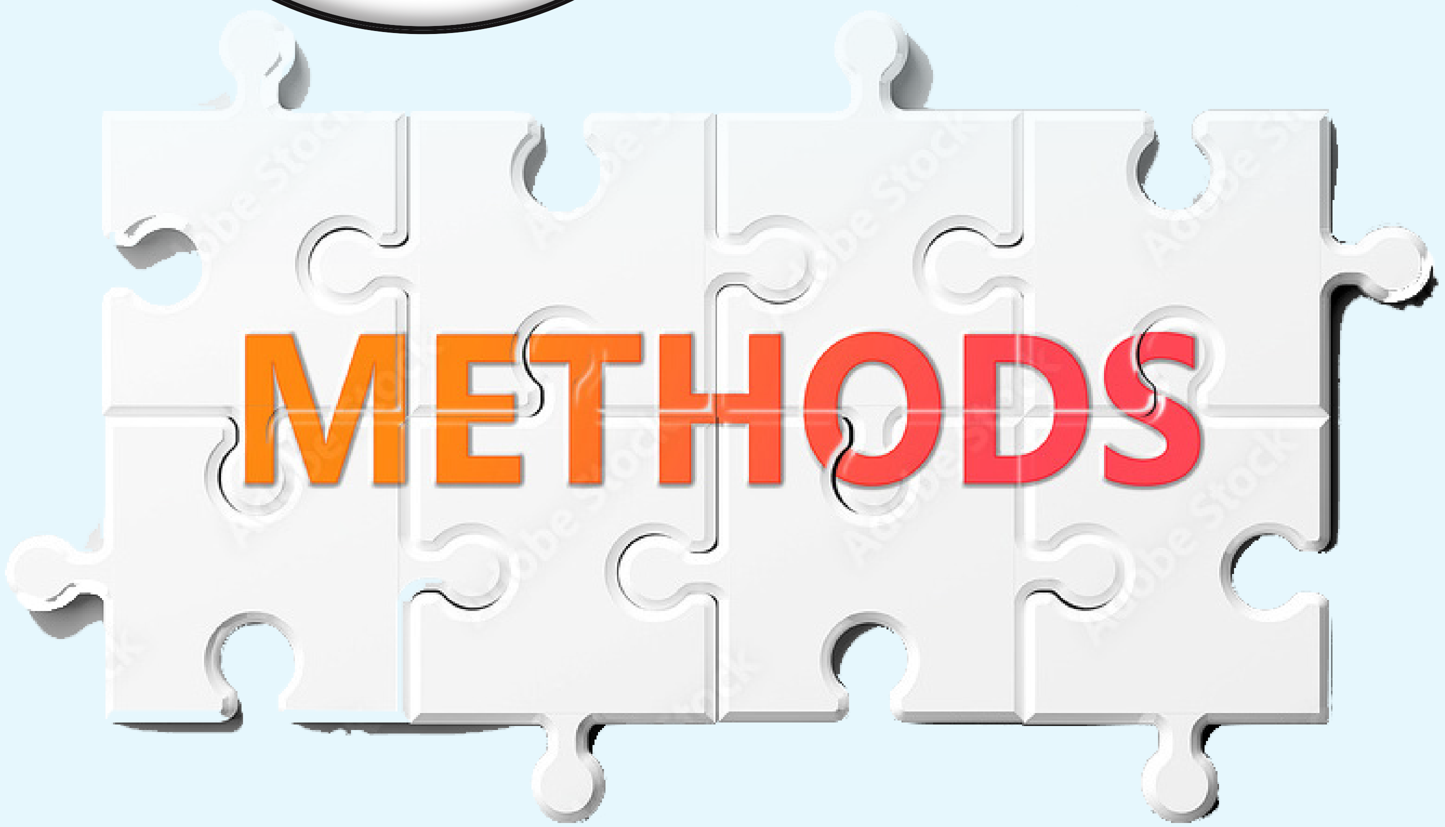
A **requirement** is a description of what the object should do, a characteristic of it, or how it should support a user. Requirements are often called constraints, criteria, wishes, demands, or goals. The requirements developed here will be refined into engineering specifications, which become part of the Problem Backlog, a high-level todo list.

Developing the design requirements is often referred to as "listening to the voice of the customer." Here this meaning is broadened to the voice of the stakeholder to emphasize that there are many different "customers" for most products.

The users are the most important stakeholders because they have the most intimate relationship with the resulting product. Their experience builds the organization's reputation for producing quality products and thus greatly influences future sales, so successful organizations work to understand how users interact with their products.

Very few people buy a consumer product without first looking at the "customer reviews" online. Customer reviews are really "user reviews" and typically focus on a product's quality. User surveys have shown that users identify a quality product as one that:

- works as it should,
- lasts a long time,
- is easy to maintain,
- looks attractive,
- incorporates the latest features, and
- is environmentally friendly.



BEST PRACTICE 4.4

Successful Product Development Organizations

Use Many Methods to Develop Design Requirements.

BEST PRACTICE KEY CONCEPTS

- There are many methods used to develop design requirements.
- Requirements need to be sufficiently developed and no further.
- Requirements are not equally important to each stakeholder.
- Requirements change over time (adding, deleting, decomposing, modifying), but only with review and care.

The Goal of This Best Practice

Organizations use many methods to understand what is needed before spending time and money to develop products. What makes this challenging is that stakeholders often cannot articulate exactly what they want, even when thinking about improvements to existing products. This gets even more challenging when a team is developing something that has not yet been envisioned by its potential stakeholders.

This best practice does not offer a crystal ball but does give the structure used by many successful organizations.

Regardless of the methods used to tease out the requirements, all are aimed at the syntax shown in Fig. 1.

As a <stakeholder> | Who

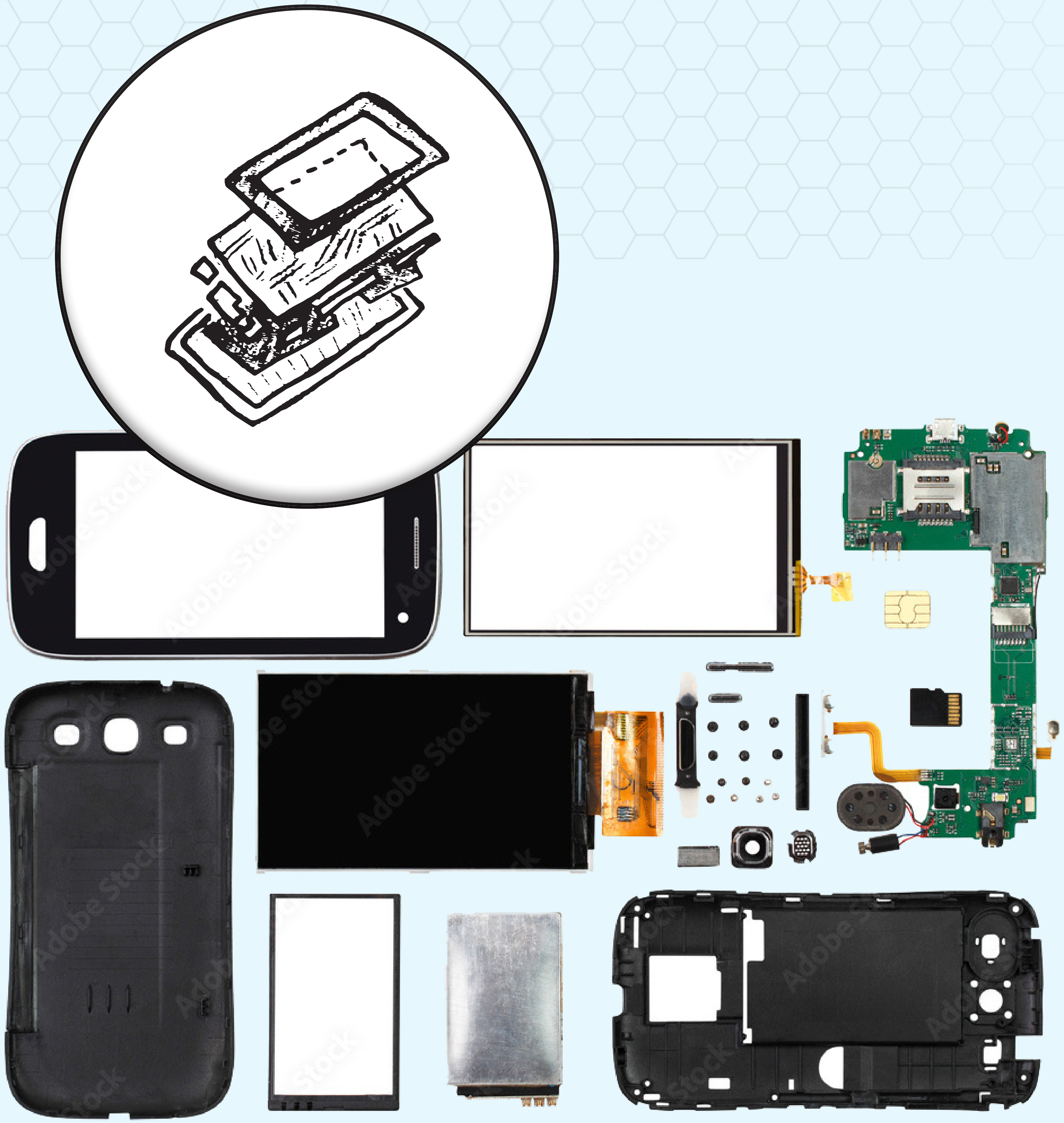
I want the <object> | What product, assembly, component or feature

to <function or have a characteristic> | What it should do

so that I <gain this benefit> | Why

Figure 1. The syntax of a requirement.

BACK TO TABLE OF CONTENTS



BEST PRACTICE 4.5

Successful Product Development Organizations

Dissect Products to Gain Product Understanding and Find Market Opportunities.

BEST PRACTICE KEY CONCEPTS

- Organizations dissect competitors' or their own products to learn about them.
- Product dissection is often called product decomposition or reverse engineering.
- The goal of dissection is to identify the individual components, how they are manufactured and assembled, the interfaces between them, and their function, as well as find market opportunities.

The Goals of This Best Practice

A TV commercial some years ago showed a team of engineers in white lab coats taking apart an SUV. The voice-over said something like, "Our products are so well engineered that our competition takes them apart to see how they work." What is not said is that the sponsor of the commercial does the same thing. All successful companies dissect their competitors' products. Sometimes, they refer to this as "product decomposition," "benchmarking," "teardown," or "reengineering." Regardless of what it is called, the goals are to answer the following:

- What components are used, how are they made, and from what material?
- How is the product assembled?
- How does the product function?

The "components" may be hardware, electronics, or code. If it is code, the list is modified into learning about the code modules, languages, function of the code, and how the code interfaces with the other components.

The dissection of competitive products (and products made within the company) can serve as a starting point whether doing a redesign, original design, or some other type of design at the product, system, or subsystem level. BMW, for example, has workshops where they deconstruct their own cars to find places to save weight and production time. Some aspects of assembly are nearly impossible to see in the CAD model but become painfully obvious when you put a wrench on the part.

It cannot be overemphasized how important decomposition is to understanding the problem. Hundreds of engineering hours

BACK TO TABLE OF CONTENTS

BEST PRACTICE **4.6**

Successful Product Development Organizations

Evolve Engineering Specifications to Measure the Design Requirements.

BEST PRACTICE KEY CONCEPTS

- Engineering specifications are refined customer requirements that are testable.
- Engineering specifications are structured as <object> <measure> <units> <target> <threshold>
- A fully refined problem backlog is composed entirely of specifications - only possible for very mature products.

The Goal of This Best Practice

Understanding the design problem is an essential foundation for a quality product. It means translating the stakeholders' requirements - the wants and needs - into a technical description of what is to be designed - the engineering specifications.

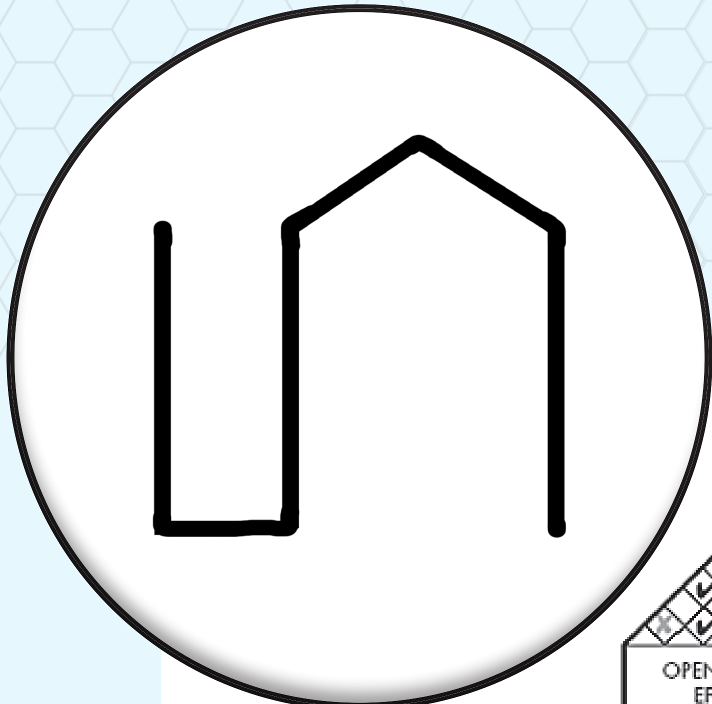
Organizations use the development of specifications as an important team communication opportunity. During this refinement, team members communicate and reach agreement about what targets they are shooting for and what constitutes a "good enough" product. Specification development also encourages communication between the team and the stakeholders as it ensures the voice of the customer as reflected in the requirements is translated into the specifications.

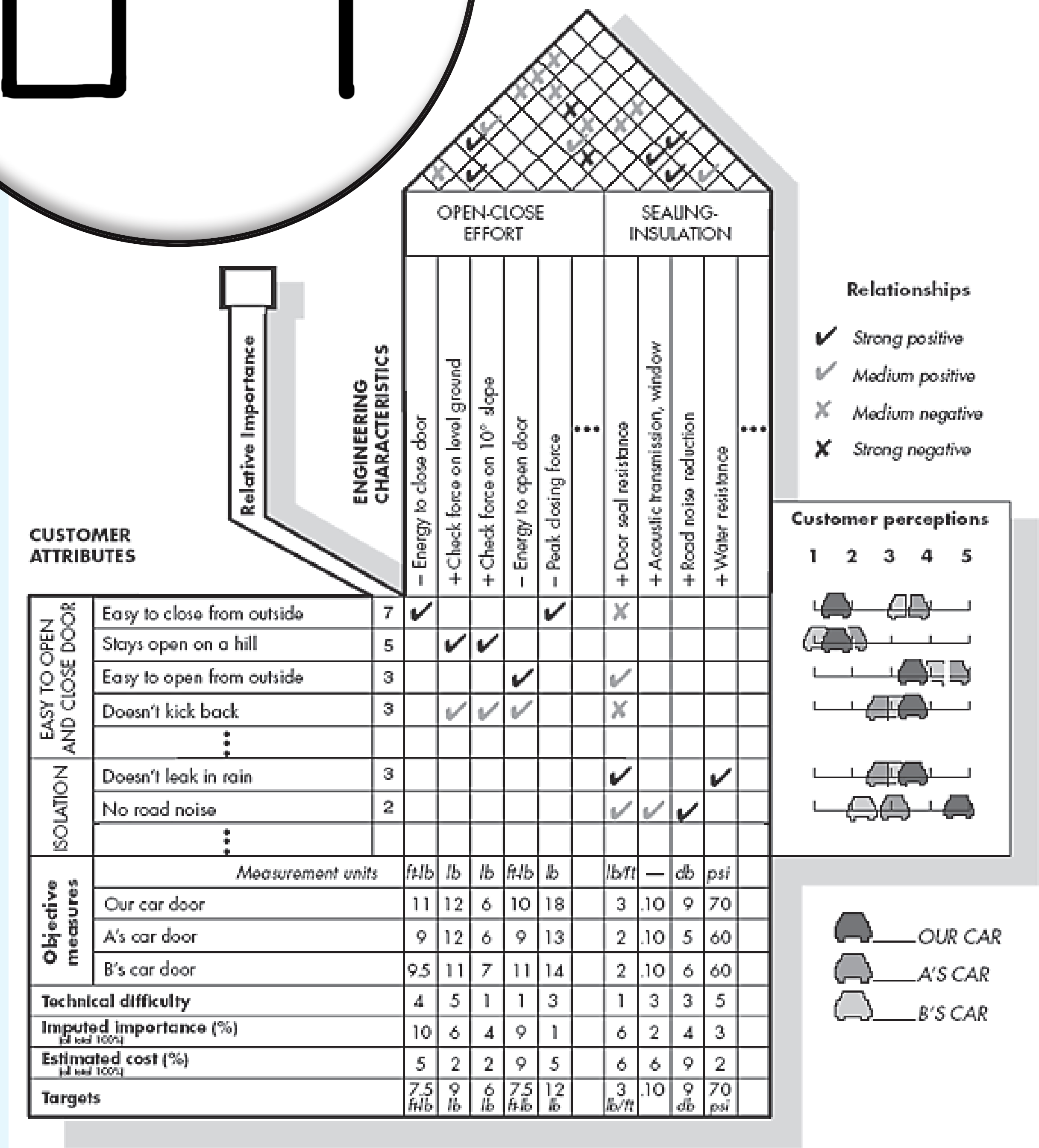
This Best Practice is an antidote for an expensive problem most organizations face: "creeping specifications" or "feature creep." **Creeping specifications** refer to the natural tendency for features included in a product to increase during the design process – e.g., "It would be nice if our new razor could also trim beards." It is estimated that fully 35% of all product development delays are caused by unmanaged changes.

For a new specification to be addressed, it must go through the same process as the original specifications or at least similar to that described here.

Kano's Model of User Satisfaction

One way to look at specifications is to examine Kano's Model of Customer Satisfaction, developed by Dr. Noriaki Kano in the early 1980s. This model helps in understanding how and why specifications exist and mature. Kano's model, as shown





BEST PRACTICE 4.7

Successful Product Development Organizations

Use Quality Function Deployment (QFD) to Develop Problem Understanding

BEST PRACTICE KEY CONCEPTS

- QFD is a major method for building problem-understanding.
- QFD is widely used in the automotive industry.
- QFD integrates stakeholders, requirements, and specification development.

The Goal of This Best Practice

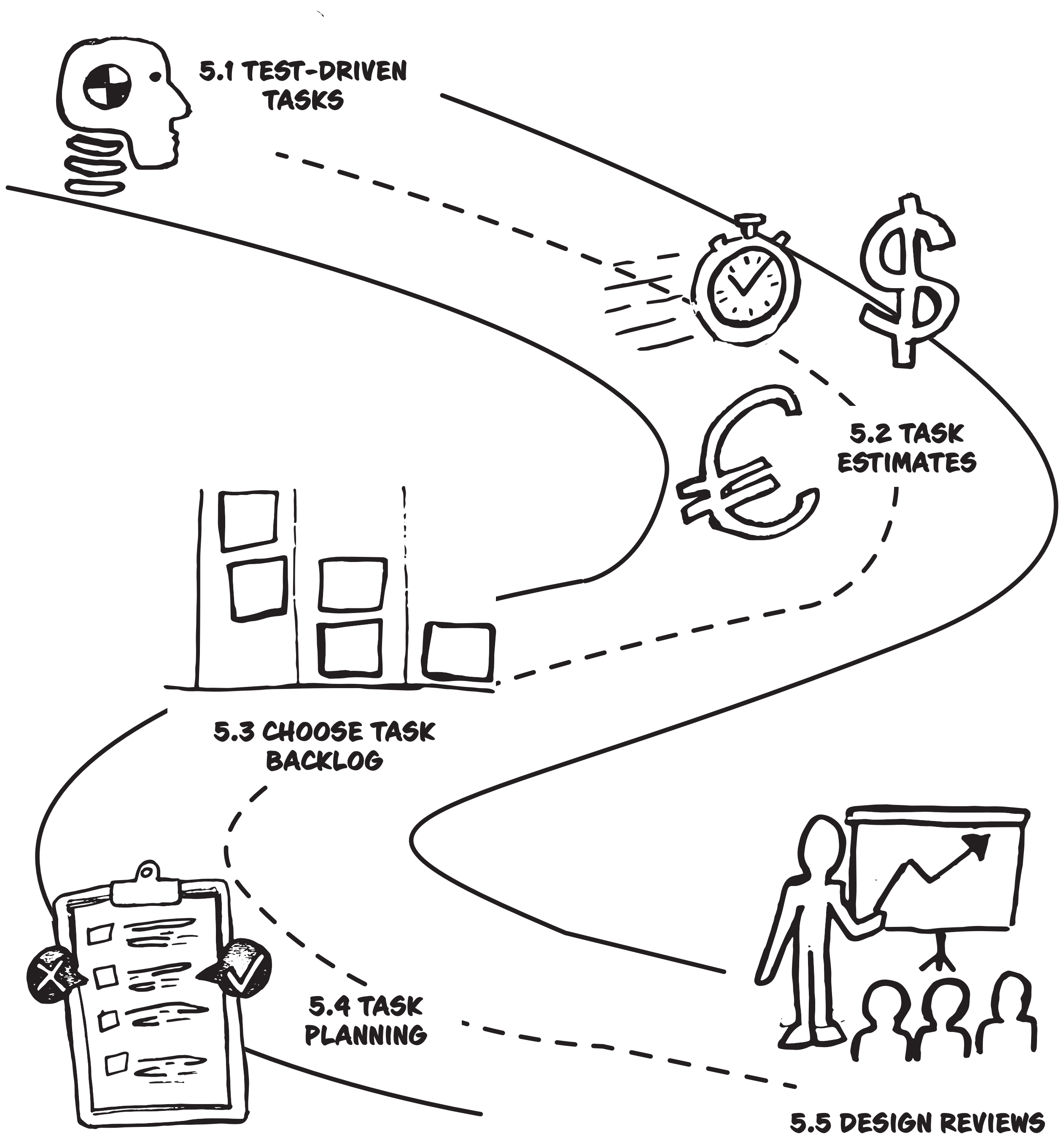
Quality Function Deployment (QFD) is a widely used method to manage the voice of the stakeholders, the requirements, and the specifications all on one sheet. It is organized to develop the major pieces of information necessary to understand the problem:

1. Identifying the stakeholders.
2. Capturing the stakeholders' requirements.
3. Understanding what is important to the stakeholders.
4. Benchmarking the competition.
5. Developing engineering specifications.
6. Relating the engineering specifications to the stakeholders' requirements.
7. Developing targets and thresholds for engineering specifications.
8. Determining the interdependence of engineering specifications.

The QFD method was developed in Japan in the mid-1970s and introduced in the United States in the late 1980s. Using this method, Toyota reduced the costs of bringing a new car model to market by over 60% and decreased development time by one-third—all while improving product quality. A survey of 150 U.S. companies shows that 69% use the QFD method and that 71% of these have begun using the method since 1990. Most companies utilize QFD with cross-functional teams of ten or fewer. Of the companies surveyed, 83% felt that the method had increased customer satisfaction and 76% indicated that it facilitated rational decisions. Whether a company uses the QFD method directly or not, successful companies develop the information in the preceding list—the component parts of the QFD method.

QFD is a well-respected method for developing requirements, measures, and targets.

BACK TO TABLE OF CONTENTS



SECTION 5.0

Planning and Managing the Design Process.

Where section 4 focused on refining the design requirements into engineering specifications, some stakeholder needs, wants, and stories may not be fully refined into engineering specifications. Also, for most problems, there are some known tasks - work to be done known from the outset. These make up the Problem Backlog, which lists potential work the team may take on. All work for the team comes from this backlog. Most organizations treat this “list” informally.

- Progress here helps answer these questions:
- Have test-driven design tasks with clear deliverables been developed (Best Practice 5.1)?
- Are the time and work estimates realistic (Best Practice 5.2)?
- Is there a clear shared task backlog of work to be done(Best Practice 5.3)?
- Is there a balance between planning and not planning (Best Practice 5.4)?
- Do design reviews learning and communication (Best Practice 5.5)?
- Is it clear what to do next?

BEST PRACTICE 5.1

Successful Product Development Organizations

Develop Test-Driven Design Tasks with Clear Deliverables.

BEST PRACTICE KEY CONCEPTS

- Tasks describe what needs to be delivered (new information/knowledge) by each unit of work in the design process.
- Test-driven development focuses task attention on the deliverables.
- Each task must have a clear owner - without a single owner, no one is responsible.
- Prototypes are developed to support task learning and testing.
- It is important to design tests to be “good enough for now.”

The Goal of This Best Practice

A plan is a proposal for a series of tasks to be done. Successful organizations focus on defining the needed tasks for a project, what order to work on them, and managing those to be put off until later.

In its most basic form, a task has the syntax –

“The <responsible party/team> will do <activity> to achieve <measurable deliverable> by <deadline>.”

The activity and a clear indication of what it means to be done are underlined in the following examples.

- A team member will research how to best grip a cup with a robotic hand and show that it is completed by presenting specs on at least five different end effectors within two weeks.
- A team member will develop a written test specification and show that it is completed by delivering it to the test facility.
- A team member will analyze the energy needed to power a prosthetic hand for one day of use and identify at least three different batteries capable of delivering the energy.

A list of the tasks to be done is called the Task Backlog. Planning adds order to this backlog.

Types of Design Tasks

Four main tasks support product development: Specification Refinement, Concept Development, Product Generation, and Product Evaluation.

Specification Refinement Tasks

Design work comprises a series of tasks with clear deliverables compared to specification targets. Thus, the more refined the specifications, the more focused the tasks and chances of success.

BACK TO TABLE OF CONTENTS

Successful Product Development Organizations

Enable Teams to Make Most Realistic Task Estimates.

BEST PRACTICE KEY CONCEPTS

- Engineers need to estimate the resources needed for projects.
- There are many methods to make estimates.
- All estimates are inaccurate – they are uncertain forecasts.

The Goal of This Best Practice

Engineers often need to estimate the product's performance, production cost, design process cost, and time to complete. Successful organizations try to make these estimates as accurate as possible.

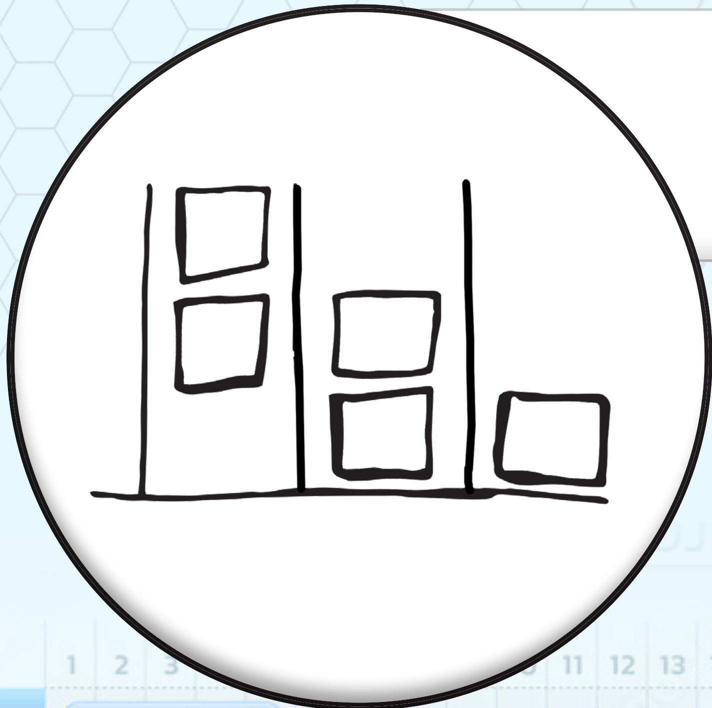
Traditionally, engineering is focused on estimating performance - how well it will work, Fig. 1. In this book, Sections 7, 8, and 9 are devoted to evaluations and tradeoffs commonly used to make the best possible performance estimates.

Beyond performance estimation, in many organizations, design engineers are responsible for estimating the final product's costs, as this is a driving requirement. Depending on the industry, some organizations have a team of professional cost estimators that estimate the money, materials, and labor required to manufacture a product, construct a building, or provide a service. Cost estimators usually exist in organizations that manufacture a narrow range of mature consumer products, such as buildings, bridges, large low volume equipment, or products for the government.

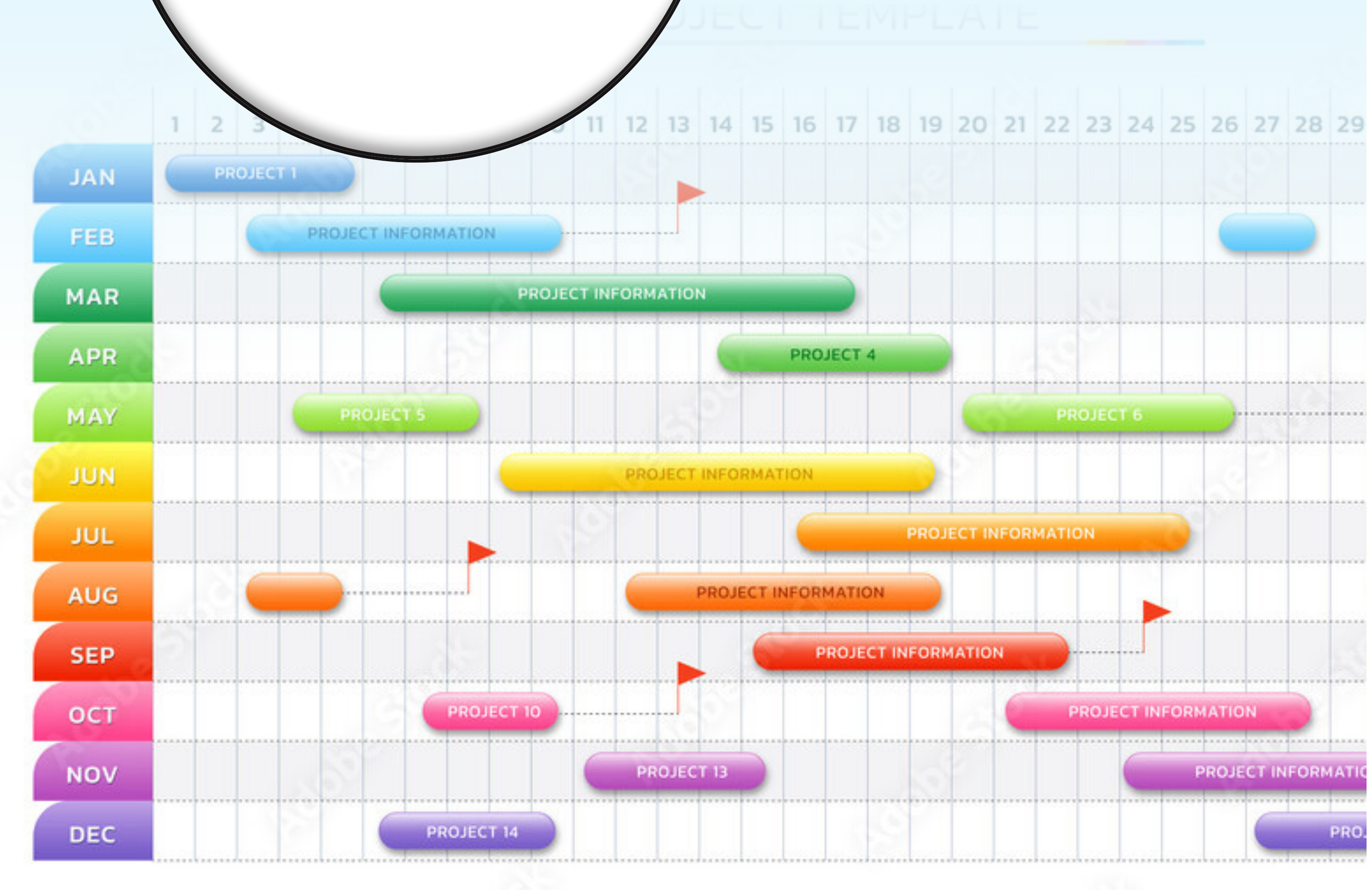
There is also the need to estimate product development time. Time estimation is a natural and ever-present tension in organizations where sales and management want the product tomorrow, and engineers always need more time to make it perfect. Thus, product designers must make the best possible estimates and be able to defend them. If management says it needs the product to be in production in the first quarter, the design team needs to be able to estimate if this is realistic. To make it possible, the product's capabilities may need reduction, or more people may need to be added to the project.

This Best Practice focuses on making the best possible resource estimates. Such estimates support major tradeoffs between time, performance, and cost

In product design, time estimation is synonymous with design cost estimation because labor is the major expense during design. This means that estimates are



> "Planning is everything,
> the plan is nothing"
>
> —DWIGHT EISENHOWER



BEST PRACTICE 5.3

Successful Product Development Organizations

Manage Task Backlogs with a Planning Model.

BEST PRACTICE KEY CONCEPTS

- Develop a plan to work on the most critical tasks first.
- A plan is a proposed sequence of tasks to develop a product that meets the design requirements and specifications.
- Linear design processes with a rigid sequence of tasks are best used for well-understood tasks.
- An Agile process provides a loose structure for the tasks and is best used for poorly understood and under-defined tasks.
- The design process followed in industry is often a mix of linear and agile.
- Too much planning leads to waste; too little planning leads to chaos.
- Successful organizations document the processes that work for them so they can be repeated consistently.

The Goal of This Best Practice

This best practice involves planning the process to address the Task Backlog. A **plan** is a proposal for action with *intent* - a detailed proposal for doing or achieving something. The intent is in terms of the tasks that need to be done and the deliverables produced at their completion. While a plan can be expressed in text, there are generally graphical **maps -** detailed diagrams for expressing the plan, such as Gantt Charts and Kanban Boards.

As soon as the planned action begins, most plans are outdated as reality and learning sets in. This was clearly articulated in a quote from General Dwight Eisenhower (34th President of the USA), who planned the D-Day invasion of continental Europe in 1945. While product design is not an invasion, uncertainty is high and planning is equally important.

The first step in planning has already been addressed: identify the tasks that need to be done. The next step is to identify the most critical tasks to do.

Task Criticality

There are always more tasks to do than there is time. Choosing what to work on next requires ordering and prioritizing tasks with particular attention to:

BACK TO TABLE OF CONTENTS

BEST PRACTICE 5.4

Successful Product Development Organizations

Design the Task Plan for Each Project.

BEST PRACTICE KEY CONCEPTS

- Projects are managed with a combination of Gantt charts and Kanban boards.
- Too much planning leads to waste, too little leads to chaos.
- Successful organizations document the processes that work for them so they can be repeated consistently.
- Plans are intended to be modified, updated, and extended.
- Change due to learning is good; reworking (fixing a mistake) is bad.

The Goal of This Best Practice

The primary goal of this Best Practice is to explore how successful organizations decide what to do next during product design. Regardless of the project's maturity or the need for a system, a component, or code, there is always too much work to clear the Problem and Task Backlogs. Part of product design is deciding where to put the available resources. While this is often seen as a management function, it is increasingly managed by the design teams themselves.

Too much planning leads to waste; too little planning leads to chaos. The amount of uncertainty determines how much can be planned. Linear process plans work well for low-uncertainty situations and ones needing tight management control; Agile works best when uncertainty is high.

The larger the project, the more the need for planning to avoid missing something important. At the same time, plans must be flexible - but not too flexible. Many successful organizations use the best of both in a hybrid linear/Agile process.

Hybrid Linear/Agile Process

An example of the use of mixed process planning is the Saab Gripen fighter aircraft. The basic architecture of the airplane is known and stable. On a physical level, it will have wings, fuselage, and landing gear; on a functional level, it will have a propulsion system and aerodynamics.

Thus, at the level of the overall airplane, the specifications are well established at the beginning of the project, and a linear plan is possible.

In fact, for the Gripen E model, 300 specifications define targets on the range, payload, and so on. These are all that are used to define the top-level tasks to design the airplane. These specifications

BACK TO TABLE OF CONTENTS

BEST PRACTICE 5.5

Successful Product Development Organizations

Make Design Reviews Part of Product Development.

BEST PRACTICE KEY CONCEPTS

- Design reviews are part of learning.
- Frequent small reviews are much more useful than occasional large reviews.
- Frequent small reviews ensure work is not conflicting or duplicated by different people.
- It is best to match the review content and frequency to the stakeholder audience.
- Results of all reviews should be well documented.

The Goal of This Best Practice

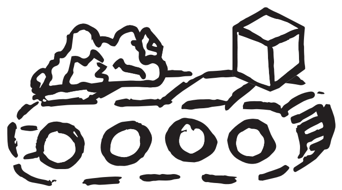
Design reviews are meetings focused on defining/understanding the problem, generating alternatives, evaluating the current solution set, making decisions, or approving the overall progress. Successful organizations understand that these meetings add value to the project team because they are not simply “updates.” Some organizations embed design reviews into a formalized, prescheduled project structure. Others treat design reviews with a “just in time” approach. In both, successful organizations provide structure to the review, focus on learning and documenting the review’s findings, and recognize the value of collaborative reviews.

Some organizations define design reviews as the “gates” in stage-gate design, where project managers decide whether the project will continue, whether resources will be allocated, or whether a new direction is needed. These are “schedule” driven design reviews. Other organizations have found success in more frequent, even weekly, reviews. This allows for more agile project coordination, bringing together the design team, external partners, and stakeholders when there is a recognized need to conduct a review. These are “event” driven design reviews. Regardless of the scheduling for the design reviews, some generalizable principles can apply to both strategies. These are the focus of this Best Practice.

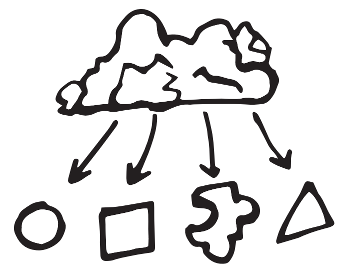
Design reviews differ from design team retrospectives. Retrospectives are by the team and for reviewing their internal design process while reviews include external partners, stakeholders, or an extended team focused on reviewing the product and clarifying its information.

Design Reviews

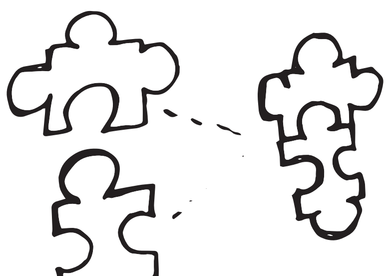
When designing a product, how do we know that the product will be successful? Does the product meet customer-defined



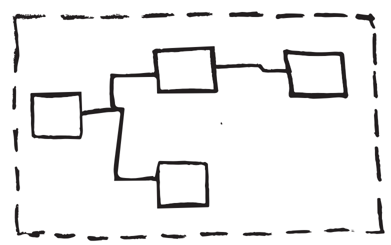
6.1 OPTIONS



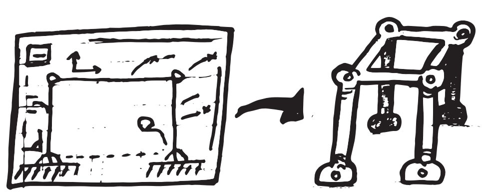
6.2 PARALLEL CONCEPTS



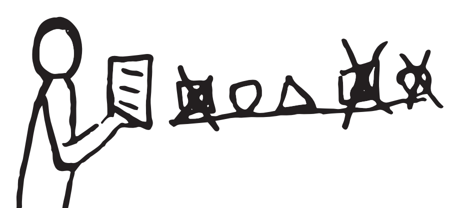
6.3 INTERFACES



6.4 FUNCTION MODELS



6.5 EMBODIMENTS



6.6 PRUNE ALTERNATIVES

BACK TO TABLE OF CONTENTS

SECTION 6.0

Alternative Generation

The best practices in this section focus on generating concepts that will lead to quality products. The bottom-line goal of this collection of best practices is to develop components and assemblies that function to meet the design requirements and specifications. Successful organizations take many steps along the way to get there effectively.

The best practices in this section answer the following questions:

- Are there optional concepts to develop?
- Can those concepts worth developing be refined in parallel?
- For each concept:
  - Are the interfaces with other objects known?
  - Is the functional flow understood?
  - Can the concepts be embodied into form?
  - How can the viable concepts be pruned for further refinement?
  - Are any resulting products, assemblies, or components worth evaluation, testing, and refinement into a product?

BEST PRACTICE 6.1

Successful Product Development Organizations

Systematically Generate Alternative Options.

BEST PRACTICE KEY CONCEPTS

- Problem or subproblem options can be generated using the same methods.
- Generating many ideas helps avoid premature design fixation.
- Generating many ideas helps in exploring the option space.
- Using systematic methods increases the likelihood of developing promising ideas.

The Goals of This Best Practice

Successful organizations recognize the importance of exploring many options to increase the likelihood of finding good design solutions. They employ several different idea generation methods rather than hoping random solutions will appear. These idea-generation methods have rules and guidelines to help foster creativity in the designer and design team. This Best Practice presents the most successful design idea generation methods.

The methods presented here can be compared based on how they support exploring the design space. First, some tools are intended to help engineers generate many ideas (quantity), assuming that good ideas will be selected and developed from the pool of concepts generated.

Second, some methods focus on generating solutions that more fully address the requirements (quality). Often, the quality of a solution is mistakenly related to either the level of detail or quality of the presentation used to describe it. This false sense of quality is because of the dangerous bias to trust concepts that are well drawn, illustrated through a solid model, or even prototyped when compared to others that are merely sketched. This particular bias can be avoided by ensuring that all concepts are represented similarly.

Third, some idea-generation methods focus on generating widely differing concepts (variety), and these methods push engineers to consider aspects they otherwise might not consider.

Fourth, some methods emphasize developing unique and unanticipated solutions (novelty). While novelty can be good, it can also be a potential trap: engineers like to play with new, different ideas, even if more traditional solutions better meet the requirements.

One essential element in many of these intuitive idea-generation methods is the concept of “provocative stimuli,” where

BACK TO TABLE OF CONTENTS

If you generate one idea, it is probably a poor one. If you generate twenty ideas, you may have a good one.

Or, alternatively

He who spends too much time developing a single concept realizes only that concept.

BEST PRACTICE 6.2

Successful Product Development Organizations

Generate and Refine Multiple Concepts in Parallel.

BEST PRACTICE KEY CONCEPTS

- All design problems have many different solutions.
- Generating many varied ideas builds confidence in the final developed solution.
- St-based or dark horse approaches help create parallel concepts.
- Use concepts to generate new understanding of the requirements and specifications.

The Goals of This Best Practice

A **concept** is an idea sufficiently developed to evaluate the physical principles that govern its behavior – how it *functions*. Confirming that a concept will operate as anticipated and (with reasonable further development) meets the requirements is a primary goal in concept development. Concepts must also be refined enough to evaluate the technologies needed to realize them, to evaluate their basic architecture (i.e., how they are put together), and to evaluate manufacturability. They can be represented as a rough sketch or flow diagram, a proof-of-concept prototype, a set of calculations, or textual notes—an abstraction of what might someday be a product. However a concept is represented, the key point is that enough detail must be present to model the performance sufficiently to ensure functionality.

Conceptual design includes systematically generating concepts that address the identified needs along with the preliminary evaluation and refinement of these concepts. Successful organizations do not rely on chance and luck to explore these ideas. Rather, they encourage design teams to generate and explore many different ideas, developing them to the point where decisions about prioritizing and investing in further development can be made as objectively as possible.

Multiple Concepts

The first concepts generated to resolve a design problem are probably not the best. Successful organizations recognize that focusing on a single idea is a potential trap for engineers (often called “design fixation”). Many successful organizations explicitly explore multiple parallel solutions to the same problem to avoid this. By encouraging and even requiring multiple options to be considered in conceptual design review meetings, organizations mitigate the risk of investing time, effort,

BEST PRACTICE 6.3

Successful Product Development Organizations

Design from Known Stable Interfaces.

BEST PRACTICE KEY CONCEPTS

- Most products have many of their components grouped into modules – functional entities.
- Modules physically and functionally interact at their interfaces.
- Designing fixed, stable interfaces allows for the independence of modules and better products.

The Goals of This Best Practice

Efficient design effort works from interfaces. This important concept is based on understanding how fixed and stable interfaces tie modules together into a product. The terms “interface” and “module” are developed in this Best Practice and apply to all engineering disciplines.

Modules, Systems, and Architectures

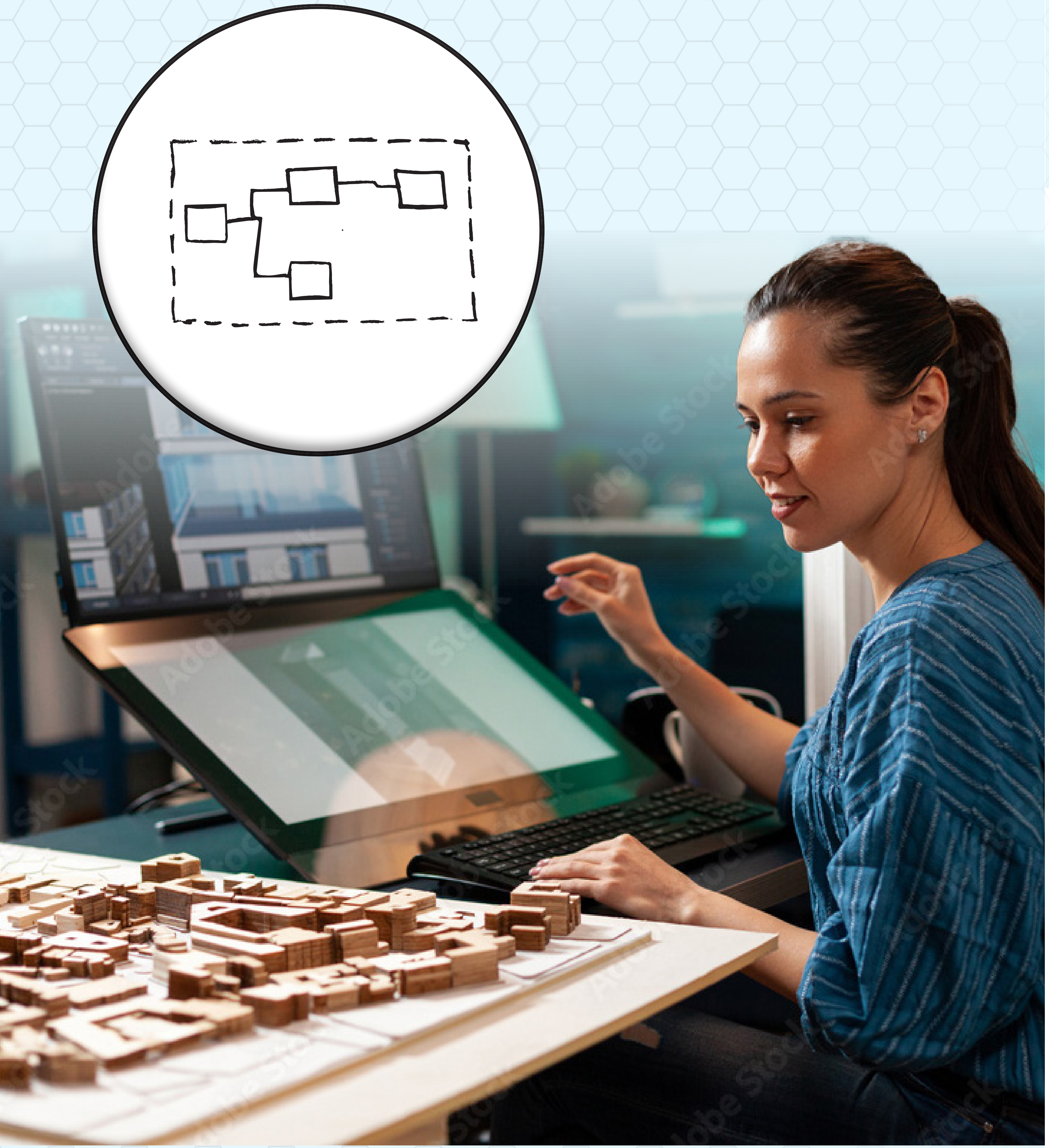
The term **module** is often used synonymously with assembly and system. A **system or sub-system** is any collection of components or assemblies grouped for function. An **assembly** is a convenient grouping of mechanical, electrical, or software components. For physical systems, assemblies are often driven by manufacturing order. One assembly must be completed before another, or one assembly may be made in one plant and a second in another, only to be brought together in a third.

For hardware, a module is an assembly that provides a specific function. Like hardware, a software module contains an independent chunk of code containing everything necessary to execute a specific function.

For example, a desktop computer is an assembly of modules. The slots on the motherboard (a module) in Fig. 1 provide a fixed, stable interface for RAM (Random Access Memory) cards (themselves modules). It has other slots for graphics cards (modules) and even a fixed interface for the processor (a module). Within the processor and the RAM are modules of code providing unique functions.

The intent differentiates assemblies and modules; manufacturing yields assemblies, and function determines modules. Frequently, the two intents overlap, creating a murky difference between them.

Another term commonly used is **architecture**, the arrangement of



BEST PRACTICE 6.4

Successful Product Development Organizations

Develop Function Models.

BEST PRACTICE KEY CONCEPTS

- Function modeling sets the foundation for the development of form and code.
- Function primarily happens at interfaces.
- Systems and the assemblies, components, and code that make them function are usually designed from interfaces - from the outside in.
- The activity of function modeling extracts new requirements, affords comparison of optional configurations, and provides a foundation for transforming to hardware and software.

The Goals of This Best Practice

The primary goal of this Best Practice is to develop a functional model as a skeleton upon which the muscles, the form of a product, are "grown." These models are black boxes that explain the functions associated with sub-systems and components. Successful organizations use an abstract function modeling approach to explore high-level product configurations.

Identifying the functions and how they relate is important in developing stable interfaces, where the sub-systems can be addressed independently with reduced influence between them. This Best Practice describes a function modeling approach and how to use it to extract new requirements, evaluate and compare optional configurations, and transform the models into hardware and software.

Effective engineers take the time to study function because:

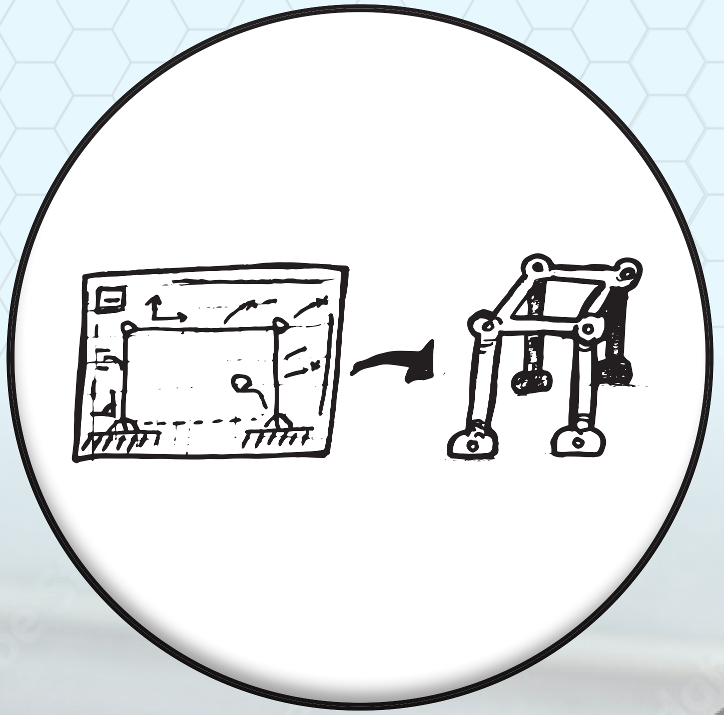
1. The bottom line for a product is that customers want products that perform as expected. Performance is the result of function, and function needs to be designed into the product from the beginning.
2. It ensures the embodiment (the form) does what it must to meet the customers' needs.
3. It drives alternative parallel embodiments.
4. It often uncovers new requirements.
5. It often uncovers assumptions.

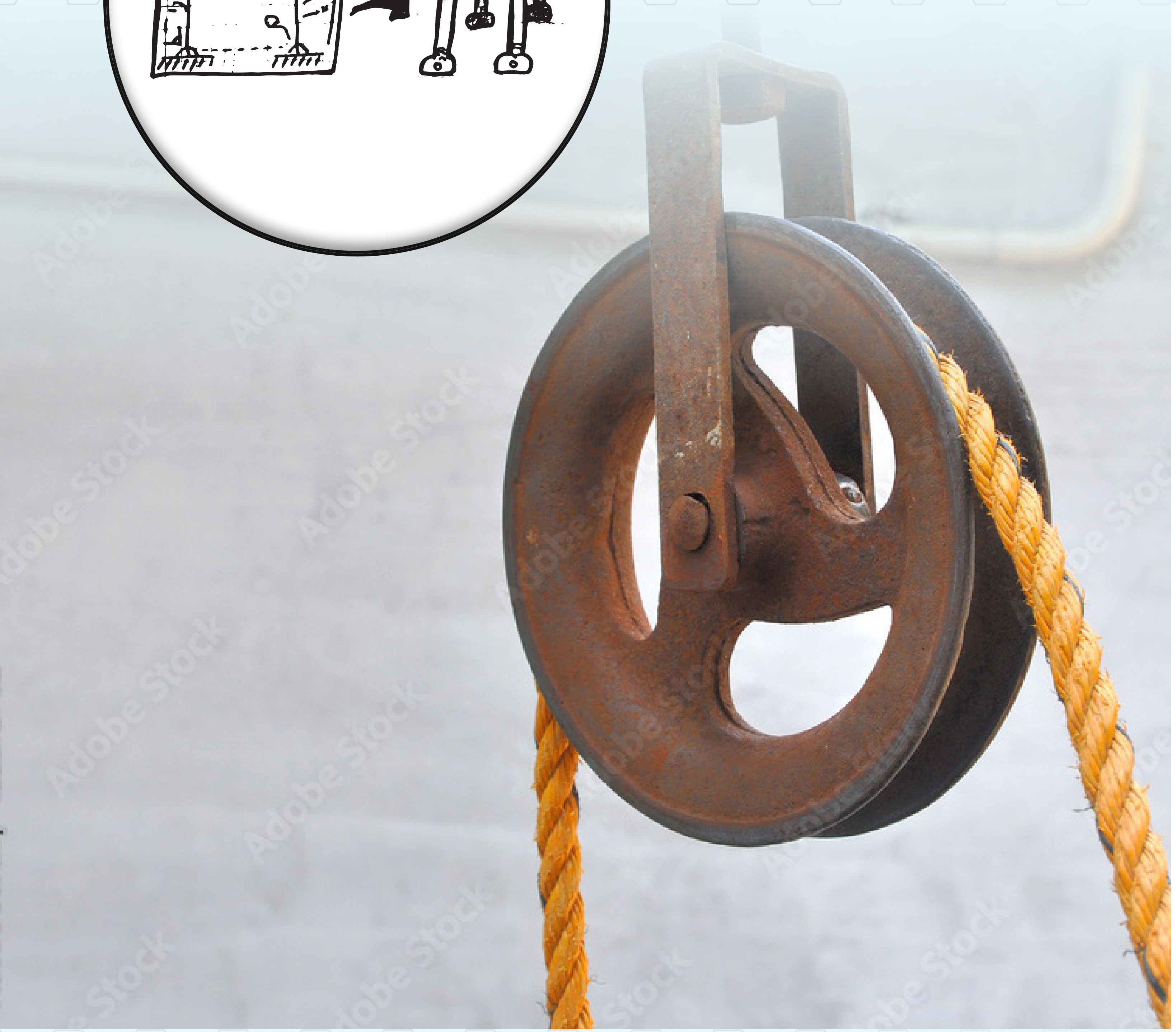
Guidelines to Building a Functional Model

Throughout the steps of building a functional model, some guidelines are universal:

- **Guideline: All functions are action verbs.** Action verbs describe the

BACK TO TABLE OF CONTENTS





Successful Product Development Organizations

Refine Concepts into Simple, Viable Products.

BEST PRACTICE KEY CONCEPTS

- Function drives interfaces while interfaces drive the form of components and assemblies.
- Design is like peeling an onion, working from the outside inward.
- Refining a product requires iteration for continuous improvement of performance.
- During iteration, there is a significant difference between refining and patching.
- Safety must be designed in a product.
- Design work is only as good as what the documentation captures.

The Goals of This Best Practice

The goal of this section is to give “form” (or embodiment) to the concepts that have been developed. **Embodiment design** focuses on parametric sizing and geometric relationships of mechanical (physical) objects (Fig. 1). For electrical engineering, it refers to the selection and sizing of electronic objects. In both disciplines, embodiment design focuses on converting abstract concepts into products. While the material here is for hardware design, much of it also applies to software development.

This Best Practice features ten guidelines for developing product components and assemblies. These guidelines are what effective engineers practice.

Effective engineers work from function to form with the requirements and specifications in mind.

Successful organizations know that if the requirements are not clear and universally understood and if the function is unrefined, time will be wasted patching any effort designing parts and assemblies.

This guideline is challenging because the concept sketches on napkins and CAD work are early commitments to the form of components and assemblies. While needed to understand the problem and communicate with teammates, they can lead to premature commitment to concepts and objects that will not perform well enough to meet requirements and specifications.

BEST PRACTICE 6.6

Successful Product Development Organizations

Systematically Prune Alternatives.

BEST PRACTICE KEY CONCEPTS

- Alternatives should be pruned, not selected.
- Pruning methods foster discussion, learning, and decisions.
- Engineers make decisions with methods that support the decision-making process.

The Goals of This Best Practice

The primary goal of this best practice is to focus on the most promising alternatives for development. Here, the term "prune" is used for choosing the most promising alternatives rather than "select." Selection implies identifying and focusing on one alternative whereas pruning suggests eliminating alternatives that exhibit little promise of being developed into a product. This difference of focus is part of the design philosophy developed by Toyota and has been adopted by many others.

The decision-making methods used for alternative pruning are:

- Pro-con analysis
- Pairwise comparison
- Decision matrix

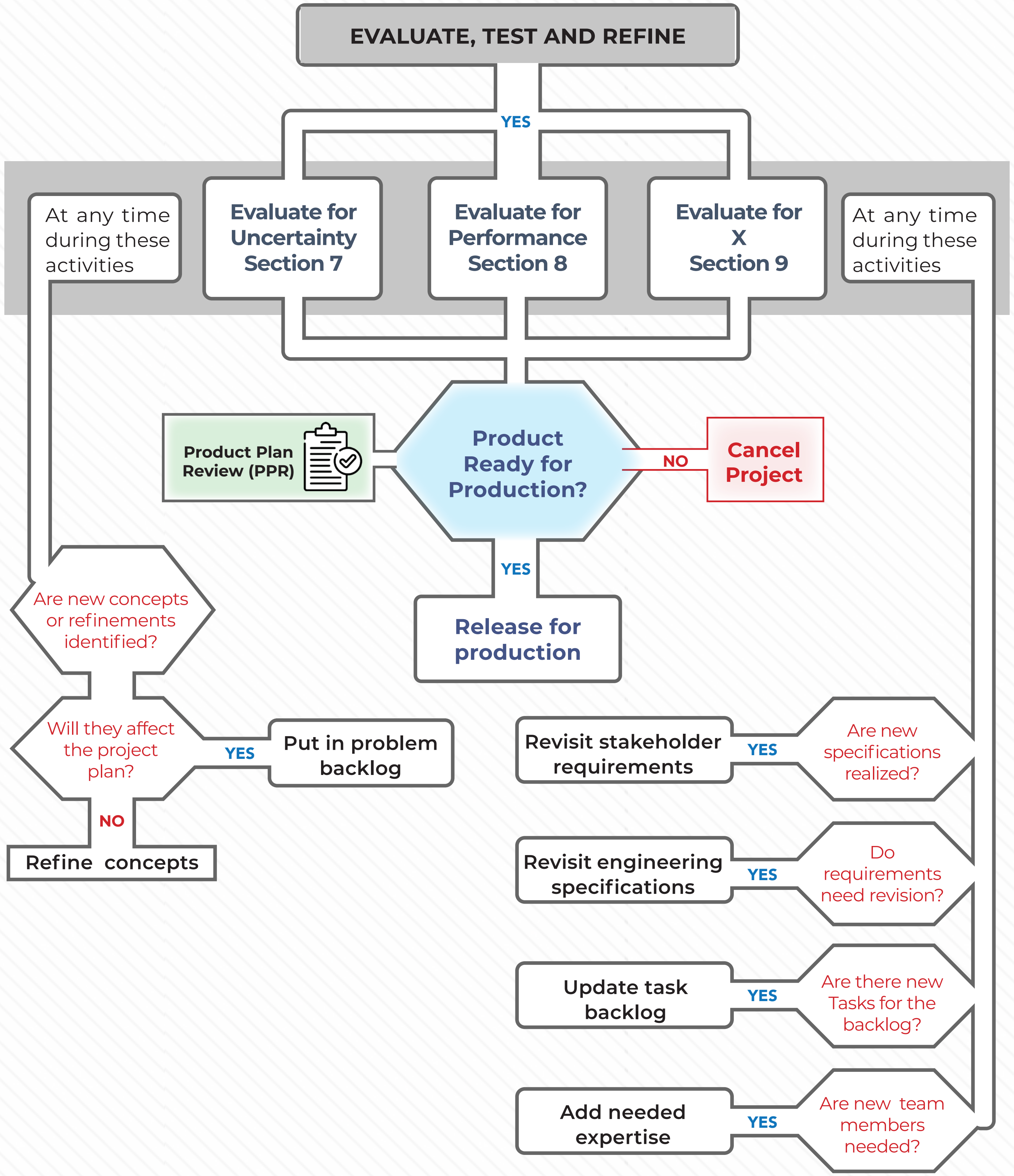
These methods support the elements shown in Fig. 1 (developed in Best Practice 2.4). While these methods can be used for many design decisions, they are specifically applied to finding the best alternatives for further development (the issue) and deciding what to do next to efficiently move the project toward a quality product. All the elements in the figure are under a cloud of VUCA+ uncertainty, with specifications, requirements, and concepts evolving as more is learned.

The decision-making activities are specifically applied to pruning the concepts using known requirements (qualitative) and specifications (quantitative), along with whatever level of evaluation is needed to focus future work on promising configurations. Further, this process reveals what to do next with better understanding of the issues, what requirements to refine, where better alternatives are needed, or where more evaluation is warranted—all through fostering communication and learning.

The challenges are:

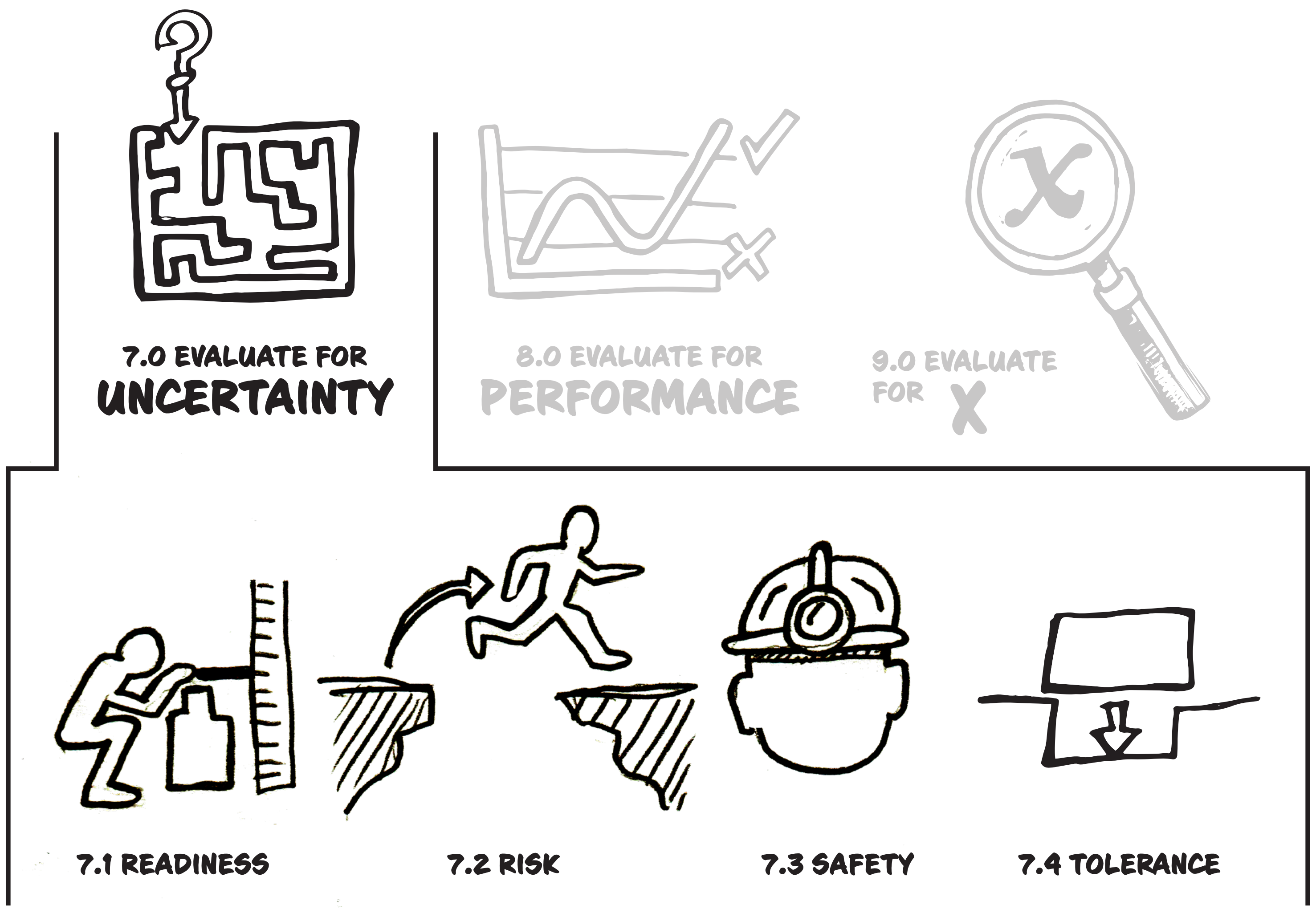
1. The measures may be qualitative (requirements), quantitative (specifications), or a mix. Qualitative measures may be in terms of "easy," "fast," or "cheap." Quantitative measures have numerical targets and thresholds.

BACK TO TABLE OF CONTENTS



Product Evaluation

This is the first of three sections about product evaluation. They are not sequential. Evaluation for uncertainty (Section 7), evaluation for performance (Section 8), and evaluation for X where X = cost, manufacture, assembly, reliability, test/maintenance, and sustainability (Section 9) are all equally important.



BACK TO TABLE OF CONTENTS

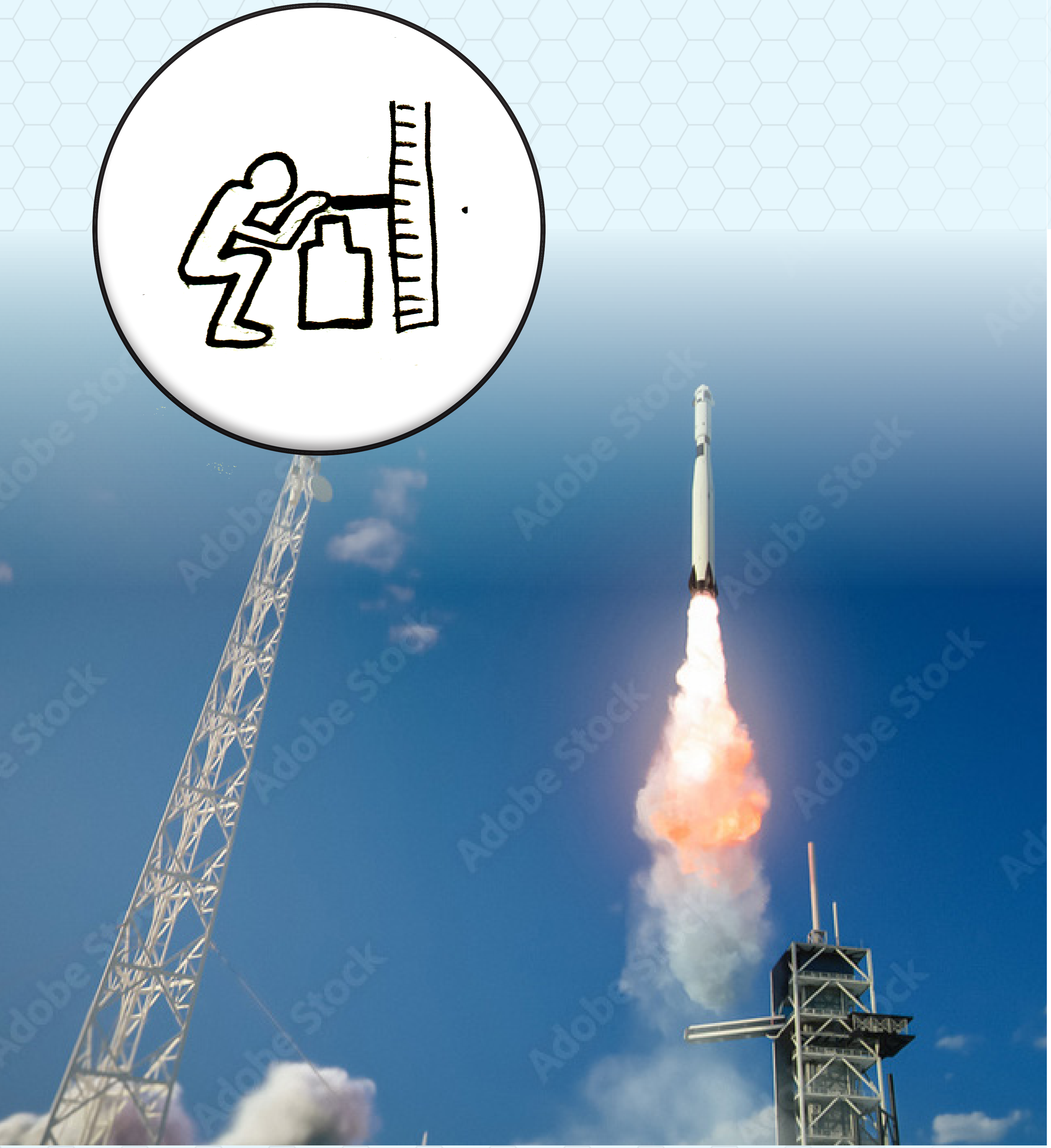
SECTION 7.0

Uncertainty Evaluation and Management

The Best Practices in this section emphasize evaluation of uncertainty and support answering the following questions:

- Are the technologies in the product ready for use?
- Are the product and project risks known?
- What factors of safety or margins should be used?
- What tolerances are consistent with needed function, fit, and manufacturing methods?

Product design is always under a VUCA+ cloud. There is always Variation, Uncertainty, Complexity, and Ambiguity, plus Incompleteness and Black Swans, as described in Best Practice 2.3. These factors cannot be eliminated, ever, and thus, their effects must be addressed as a part of the design process.



Successful Product Development Organizations

Assess Each Technology's Readiness.

BEST PRACTICE KEY CONCEPTS

- Every technology used in a product has a “readiness,” a measure of how easily it can be integrated into a product.
- Successful organizations are aware of the readiness of each technology proposed and used.
- The technology readiness assessment process yields much information about the state of product development and provides a structure for communication.

The Goal of This Best Practice

Products are, at their core, a marriage of technologies. Some technologies provide function while others embody hardware, systems, and software. To deliver a product, all its technologies must be sufficiently mature. This Best Practice consists of seven questions to be asked of each technology to assess their maturity during the design process. These questions help determine each technology’s Design Technology Readiness Level (DTRL). The answers to these questions give guidance about what needs to be done to finalize the product, improve communication within the design team, and help to communicate the

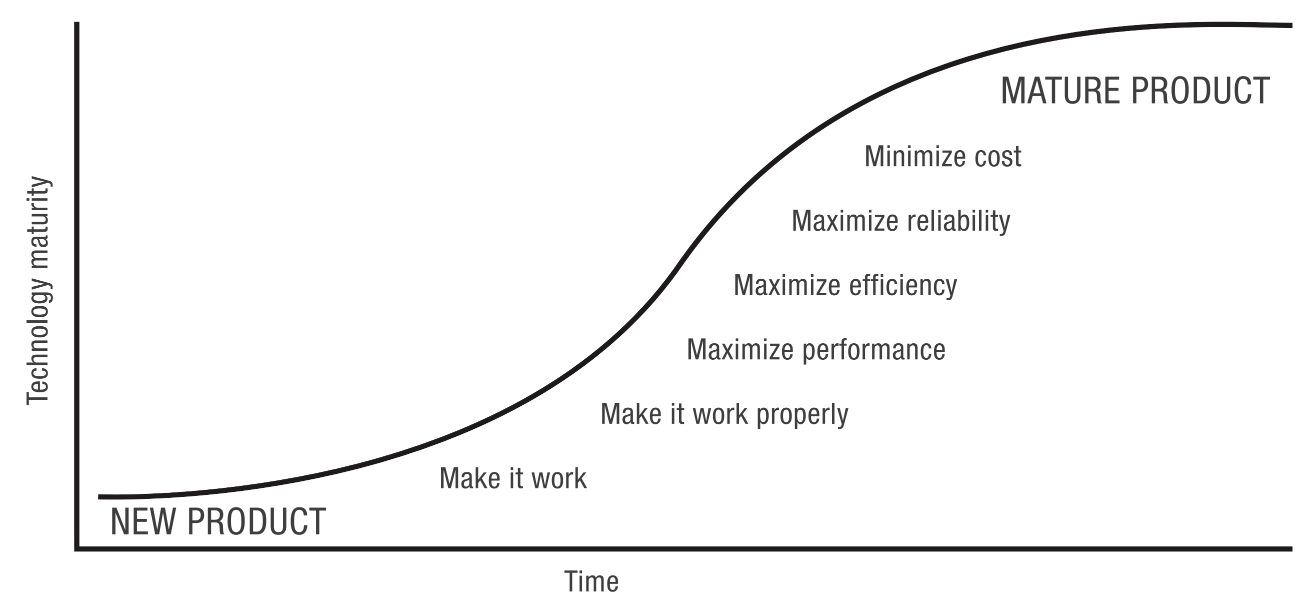


Figure 1. The Technology Maturity curve.

BEST PRACTICE 7.2

Successful Product Development Organizations

Begin Risk Assessment During Conceptual Design.

BEST PRACTICE KEY CONCEPTS

- Risk is the likelihood of something happening times its consequence.
- Risk is caused by uncertainty and thus cannot be fully mitigated.
- Engineers need to be aware of product, project, and decision risks so they know what to spend time addressing.
- A product failure can result from poor risk management during the design process.
- Products liability is the area of law where organizations and individuals are held responsible for product failures.

The Goal of This Best Practice

Organizations seek to minimize the risk associated with product development. Sometimes, this desire is stated explicitly, but often goes unsaid. Usually, engineers are concerned only with **product risk**—the risk that the product fails and potentially hurts someone or something. This view is too narrow—beyond the risk of product failure, there is the risk of the project failing to meet its goals, being behind schedule, or going over budget. Further, there is the risk that poor decisions will affect the product, the project, or both. This Best Practice explores how successful organizations manage product, project, and decision risk.

Formally, ***risk** is an expected value, a probability that combines the likelihood of something happening multiplied by the consequences of it happening*. Thus, risk depends on the answer to three questions:

1. What can go wrong?
2. How likely is it to happen?
3. What are the consequences of it happening?

These three questions are central regardless of the product, project, or decision focus.

Risk is a direct function of uncertainty. Some uncertainty is just part of nature and cannot be controlled (the weather, people's use of the product, new unforeseen laws or regulations, etc.) or only controlled at great expense and difficulty. Much of the uncertainty during conceptual design is due to a lack of knowledge about the problem and the evolving product. A product with known or insignificant risk is possible if everything is known precisely. Unfortunately, incomplete knowledge, low-

BEST PRACTICE **7.3**

Successful Product Development Organizations

Use Factors of Safety and Margins as Design Variables.

BEST PRACTICE KEY CONCEPTS

- Factors of safety and margins account for uncertainty during design to reduce product risk or give room for change.
- A factor of safety is the ratio of what is allowed to what is applied.
- Margins are the difference between the allowed and the applied and are equal to the factor of safety minus one.
- Factors of safety and margins are commonly used in mechanical design.
- Margins are used in electrical engineering and computer science.

The Goal of This Best Practice

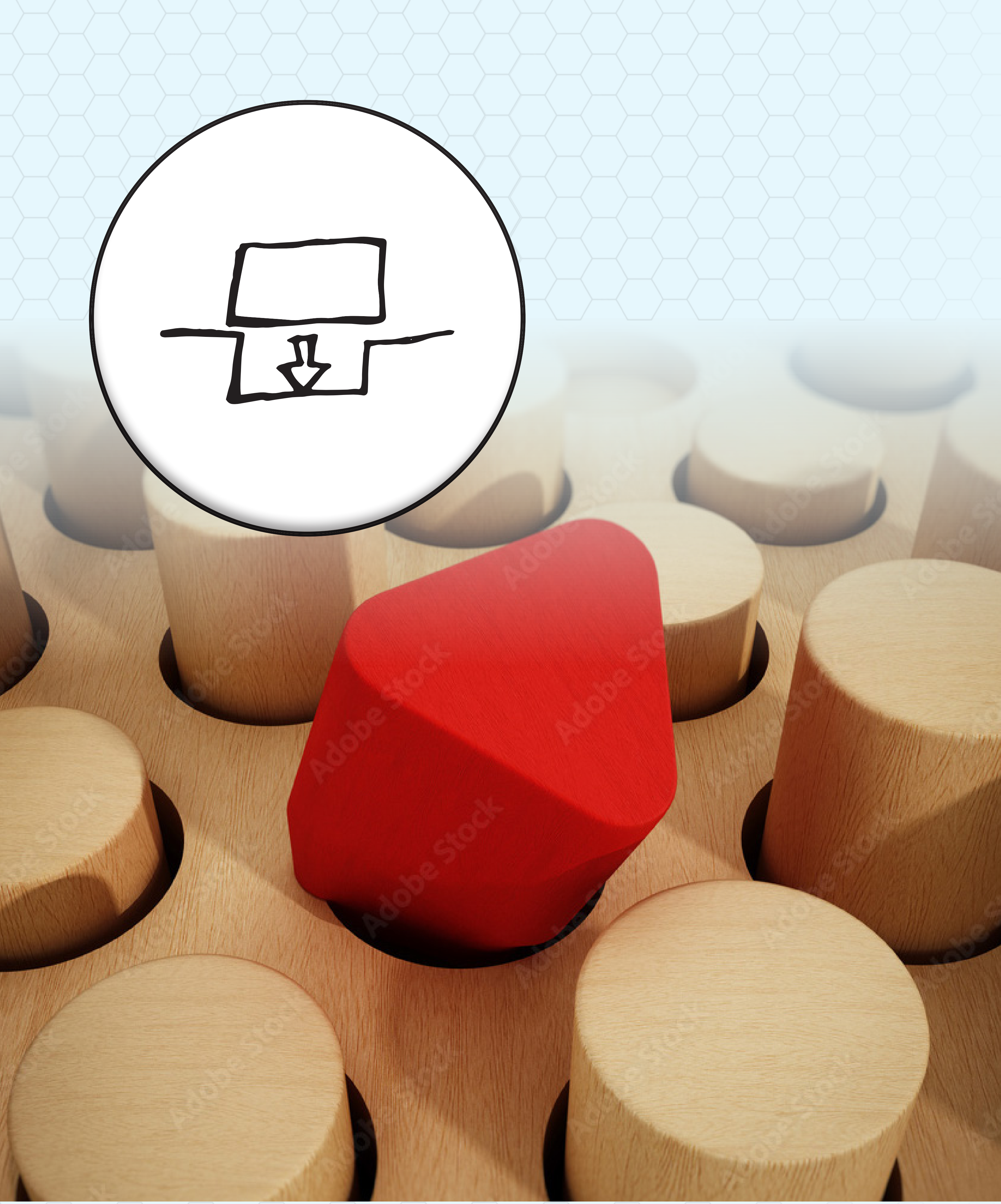
In a perfect world, a component specified to be 50 mm long would be exactly 50 mm long. A resistor with black, red, yellow, and green bands would be exactly 20,000 ohms. A material would have a yield strength of exactly 100 MPa. However, in the real world, none of these are true. All the dimensions, loadings, physical properties, and the ability to analyze the part's behavior are uncertain, and this uncertainty is compounded in systems and assemblies.

To accommodate for the uncertainties, engineers use factors of safety and margins to over-design systems. Instead of designing for a load of 20 N, they may design for 40 N. If a cooling system needs a flow rate of 7 LPM (Liters Per Minute), they may specify 14 LPM. These over-designs give a factor of safety on the conservative side but at a cost in weight, price, and other factors. Factors of safety and margins are buffers against failure and are used for determining the amount of overdesign needed for safety.

This Best Practice spells out how successful organizations use margins and factors of safety as design variables. In fact, engineers often include these factors in design specifications. For example, when designing an airplane, the Federal Aviation Regulations (FARs) prescribe load factors from -3.0 gs to 6.0 gs for acrobatic airplanes.

Wilbur Wright stated in one letter that he and Orville used a factor of 5 for their early machines. These are often called g-loads. An example of how they are used is, a 4,000 N airplane must be designed to take 24,000 N (6.0gs) when the controls are pulled back hard at high speed. These load factors are factors of safety where an airframe must demonstrate that it can survive this higher loading and withstand the stresses of acrobatic flight.

BACK TO TABLE OF CONTENTS



BEST PRACTICE 7.4

Successful Product Development Organizations

Develop Tolerances Consistent with Needed Function, Fit, and Manufacturing Methods.

Best Practice Key Concepts

- Costs generally increase exponentially with tighter tolerances on mechanical and electronic components.
- In mechanical assemblies and electronic circuits, tolerances stack up to affect fit and function.
- There are three ways to calculate stack-up: additive (worst case), statistical, and Monte Carlo.

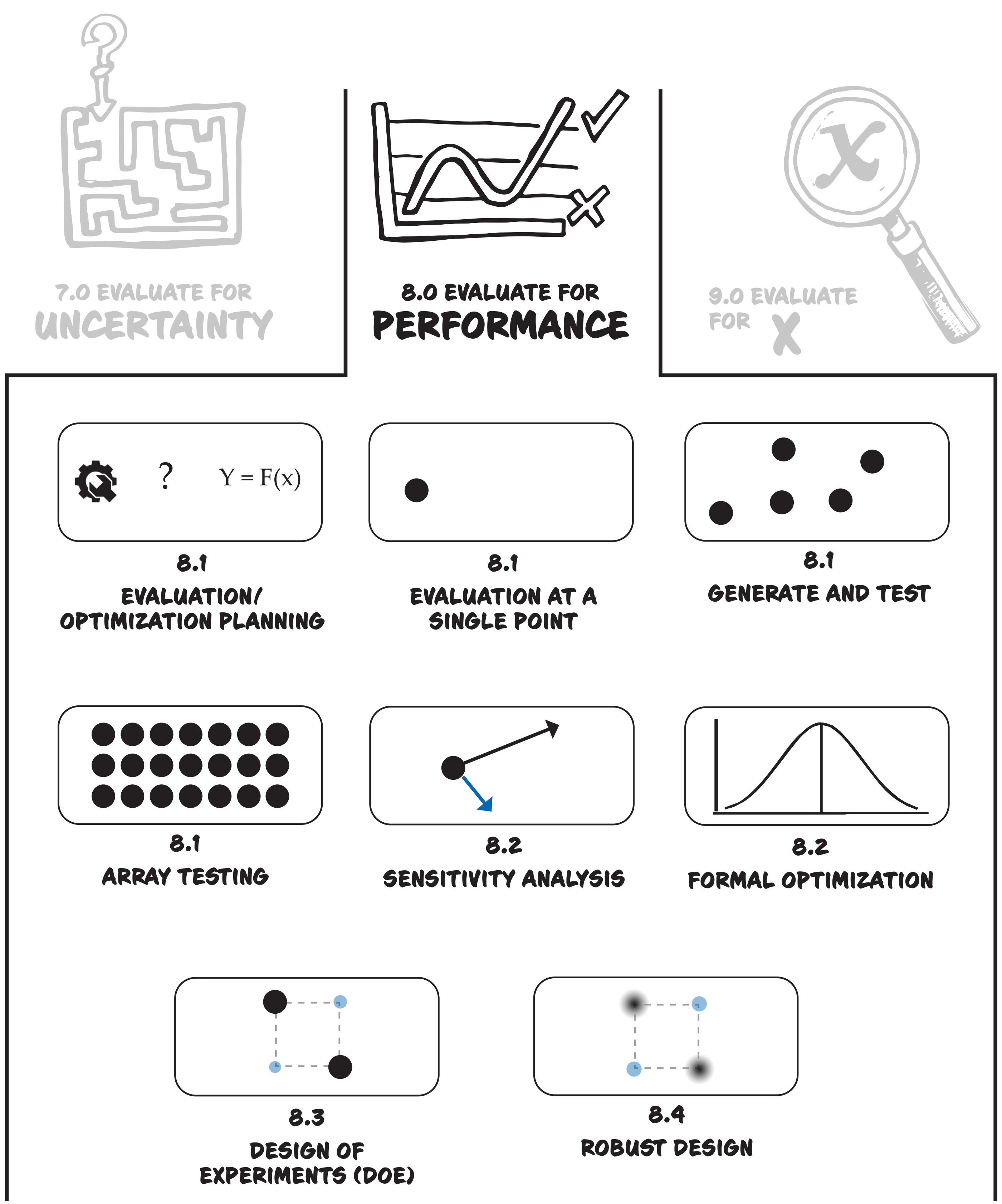
The Goal of This Best Practice

Successful product development organizations pay close attention to tolerances as they affect product function, manufacturing cost, and assembly ease.

There are three ways the term "tolerance" is used in industry. For the design engineer, **tolerance** is the allowable variation that a component or assembly can tolerate while still functioning and remaining easy to assemble. Often, this is referred to as a concern for "fit and function." During design, tolerances are determined for dimensions and other physical properties like resistance, opacity, and tensile strength. An important part of developing sufficient tolerances is communicating them downstream through documentation and notation on drawings.

The second use of "tolerance" is by manufacturing engineers. They use the tolerances communicated to them by designers to determine which technologies, machines, and tools to manufacture the components. If a design engineer communicates a very tight tolerance to manufacturing for a dimension that does not affect the fit or function it usually leads to undue manufacturing costs, as will be seen.

The third use of "tolerance" is for inspection. Typically, a Quality Control (QC) or Quality Assurance (QA) engineer will inspect a sampling of the parts produced to see if they meet the tolerances specified by the design engineer. The inspection process is referred to as "quality control," and the terms "QA" and "QC" generally mean the same thing and are often noted as "QA/QC." QA/QC inspectors assume that the design



BACK TO TABLE OF CONTENTS

SECTION 8.0

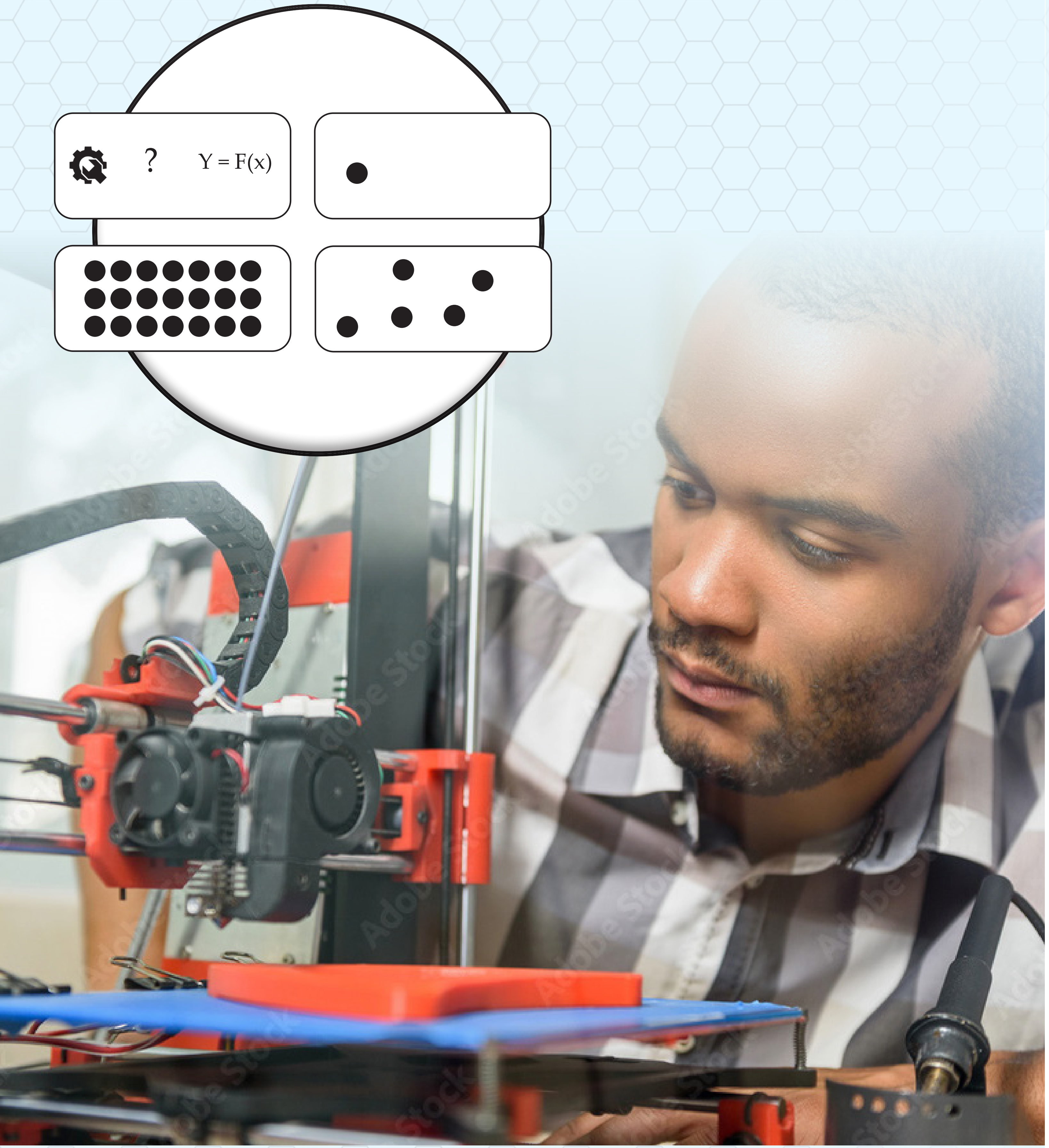
Best Practices for Product Performance Evaluation and Optimization

The Best Practices in this section answer the following primary questions:

- How can we test it?
- Does it work?
- Does it work as it should?
- Can its performance be improved?
- What is the best method to test performance and optimize it?
- What is the effect of uncertainties on performance?

These questions require evaluating performance by physically testing entities, studying them analytically, or combining both. Whether using physical testing or analysis, there are many approaches to answering the questions, some more effective than others.

Evaluation is the cornerstone of an engineering education. Most courses focus on methods to evaluate structures, circuits, thermal and control systems, and mechanisms, amongst other disciplines. None of those are covered in this book. Rather, the material here is how to apply those analysis and testing methods to answer the above questions.



BEST PRACTICE 8.1

Successful Product Development Organizations

Have a Clear Performance Evaluation and Optimization Plan.

BEST PRACTICE KEY CONCEPTS

- Evaluation determines how well a concept works and guides its optimization.
- Evaluation is the culmination of Test-Driven-Development (TDD).
- Since the design specifications targets and thresholds are numerical values, TDD requires the product to be refined sufficiently to make numerical comparisons to them physically or analytically.
- Care must be taken with analysis tools: the results are only as good as the user's knowledge of the tools, the values input, and the underlying physics.

The Goals of This Best Practice

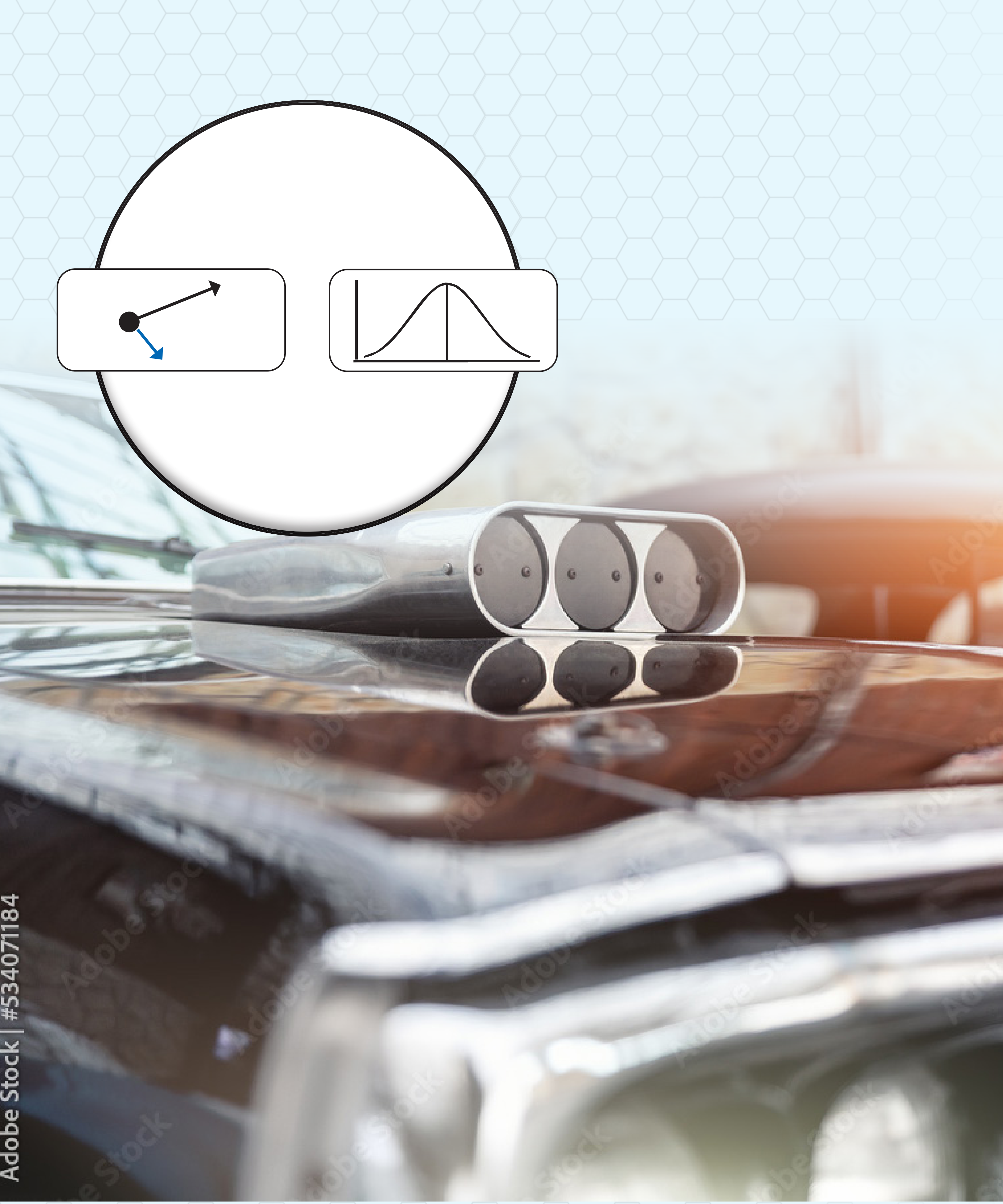
The goal of evaluation is continued learning. Only through analysis and physical modeling can engineers understand the true nature of concepts, the physics underlying them, and a concept's ability to meet specifications. No product springs complete and refined without extensive evaluation and subsequent refinement.

Beyond testing relative to the specifications, evaluation often leads to discovering new or refined requirements. Also, evaluation always raises new, unforeseen issues. For instance, in developing X, the engineers started with specifications for Y and Z. In building their first cardboard prototype, they realized that X could not work without adding U, which needed new requirements (targets and thresholds) for the system to perform.

Evaluation can be expensive. Time and costs are associated with developing and running analytical simulations, added to the expense of building and testing physical prototypes. Sound engineering involves learning and achieving quality with minimal use of expensive resources. However, successful organizations know that they must "fail early and fail often" to develop quality products. So, there is a balance between the benefit of evaluation and the cost of it, both in terms of dollars and time. Evaluation planning is all about making the cost/benefit decisions.

What Does "Evaluation" Really Mean?

Evaluation is the act of assessing concepts relative to the engineering specifications. At its most basic, evaluation answers the question, "Does the concept work?" Where "work" can only be assessed by how well the concept performs relative to engineering specifications. At its most sophisticated, evaluation guides



BEST PRACTICE 8.2

Successful Product Development Organizations

Optimize their products to improve performance.

BEST PRACTICE KEY CONCEPTS

- Trade-offs are used to identify the most balanced technical solutions among a set of proposed viable solutions.
- Sensitivity analysis is one technique for managing trade-offs and improving performance.
- During design, “optimize” usually refers to the informal methodology used to make a system as effective as possible. It is seldom possible to use formal mathematical procedures (such as finding the maximum of a function).

The Goals of This Best Practice

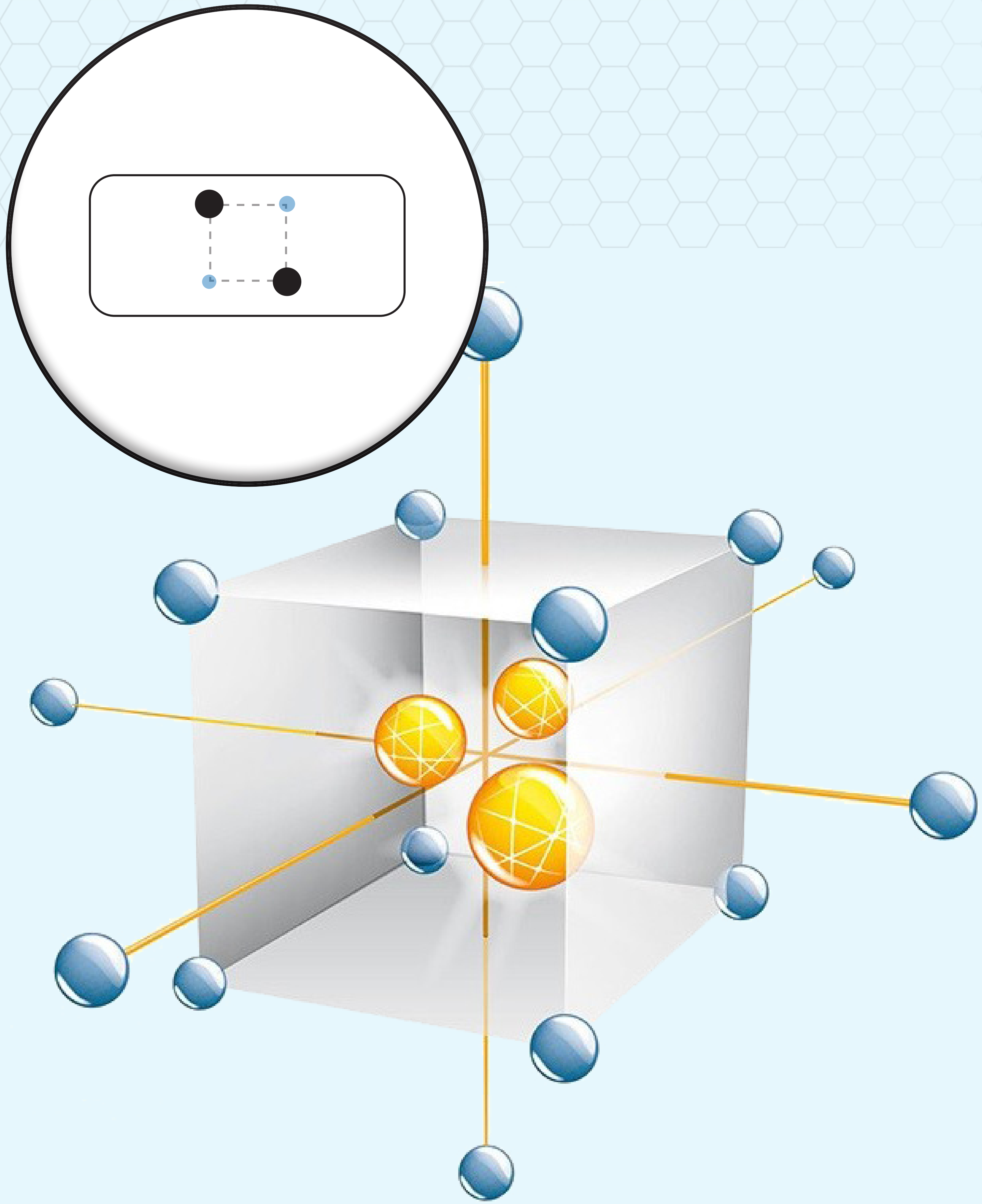
This Best Practice is titled “optimization,” but only a small part is about the formal methods of mathematically modeling a system and finding where the first derivative equals zero. This is for four reasons:

- For many design problems, no set of equations adequately describes the systems.
- There are generally many important measures. There are no single or simple combinations of KPIs.
- Finding an optimum is generally not necessary, just a good solution to meet the requirements.
- Formal optimization methods can not include uncertainty, and high uncertainty generally characterizes design problems.

When referring to mathematical, traditional optimization, it will be written with a capital “O,” where optimization (little “o”) refers to other methods that drive for improved performance.

Here, **optimization** is an iterative evaluation technique with the goal of maximizing performance. While textbook problems are designed with one right evaluation answer, design problems are generally generate-evaluate-change-iterate. Successful organizations build their evaluations and testing to drive iteration and move their products’ performance toward an optimized solution. Evaluation is a key step in optimization.

This Best Practice and the next two all leverage evaluation results to improve product performance as shown in Fig. 1. The methods in these best practices can be loosely called “optimization.”



BEST PRACTICE 8.3

Successful Product Development Organizations

Use Design of Experiments (DOE) to Support Product Development.

BEST PRACTICE KEY CONCEPTS

- Design of Experiments (DOE) methods can be used to understand the effect of variables on the key performance indicators (KPIs).
- DOE is a powerful design optimization method.
- DOE is both the exploration of the design space and is used to find “nominal-is-best,” “more-is-better,” or “less-is-better” performance for KPIs.
- DOE “experiments” can be physical or analytical.
- DOE makes use of the Analysis of Means (ANOM).

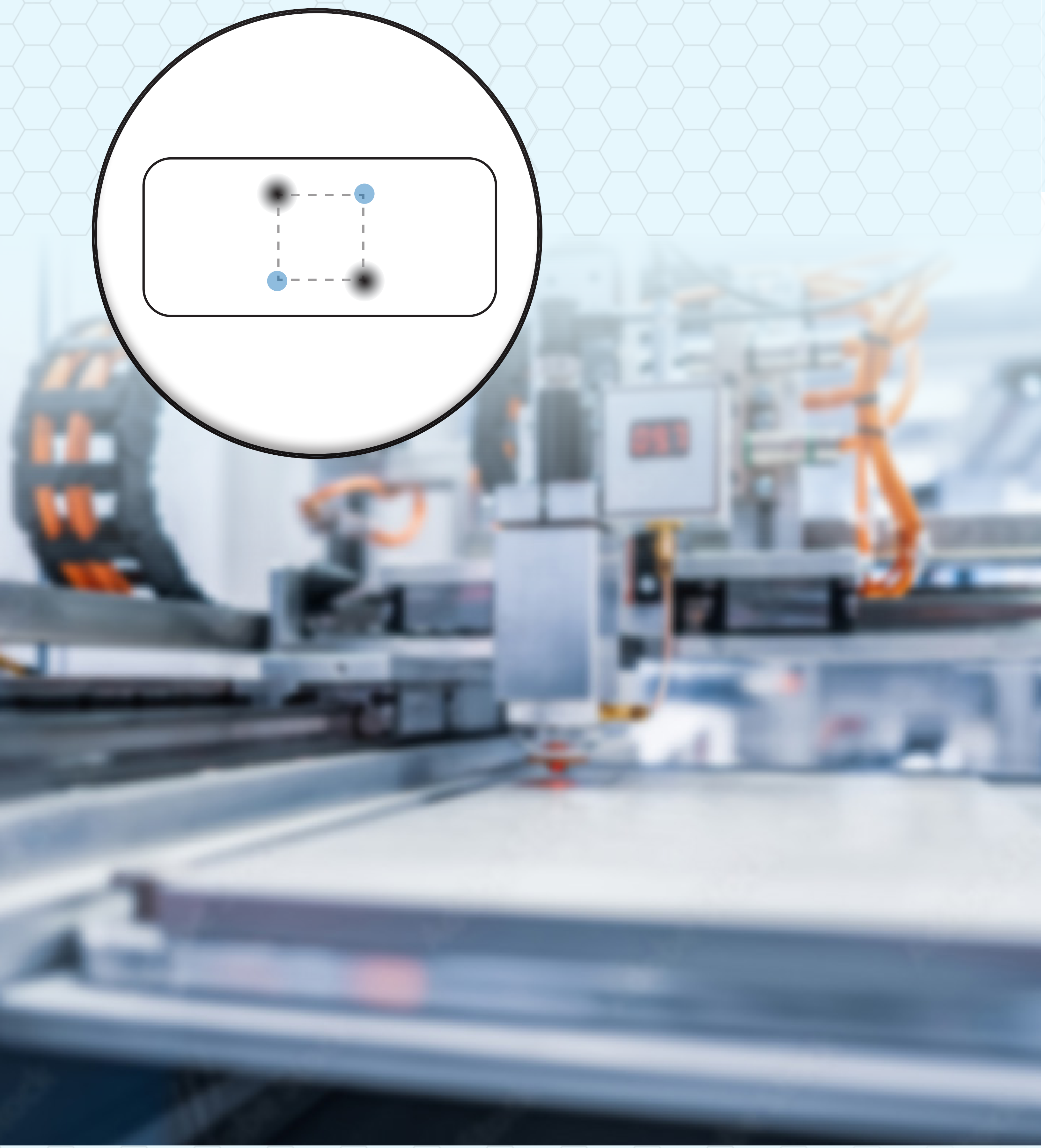
The Goal of This Best Practice

Most design problems have:

- Multiple key performance indicators (KPIs). It may be important, for example, to be fast and controllable and reliable, as well as inexpensive.
- Many design variables and their effect on the KPIs may not be known. Further, the variables may be continuous, discrete, or even qualitative. The goal is to find the set of these variables that maximize (minimize) KPIs or lead to specific target values.
- There may be a weak or potentially unknown relationship between the variables and the KPIs. There may be known equations with the KPIs as a function of the variables, but most likely, there are not.
- A noisy design space. The variables and relationships may change from unit to unit (variations in manufacturing), change with age, or be affected by the environment. The VUCA+ uncertainties may cause some of this noise.

Traditional Optimization cannot address any of these situations, but Design of Experiments (DOE) methods can. Successful organizations recognize this and use these methods.

This Best Practice introduces DOE methods



BEST PRACTICE 8.4

Successful Product Development Organizations

Develop Robust Products.

BEST PRACTICE KEY CONCEPTS

- A robust design is insensitive to noise. Noise is what the designer cannot or chooses not to control.
- Noise is uncertainty typically caused by manufacturing variations, use, aging, and environmental conditions.
- Uncertainty due to noise makes the design space fuzzy.
- Robustness can be designed into products.
- Design of Experiments (DOE) methods can be extended to produce robust designs.
- Robust Design experiments can be physical or analytical.

The Goals of This Best Practice

The concept of "robust design" originated with Dr. Genechi Taguchi in Japan shortly after WWII. Robust Design works to reduce the effect of noise before bringing the performance onto target. This can best be explained through a simple analogy, as shown by the targets in Fig. 1.

An archer is shooting at a target. Many noises are affecting the ability to repeatedly hit the bullseye: the wind (an environmental noise), arrow variation (a manufacturing noise), lack of muscle control (another environmental noise), and so on.

To get repeated bullseyes, there are two things they need to manage. First, the shots must be consistent (or precise), all bunched together with low variance. Second, they must be on target, hitting the bullseye accurately.

Traditional design methods focus on accuracy, getting a design to meet the requirements (be on the bullseye), and then considering the effect of noise (reduce the variance). Taguchi turned this around, noting that it is harder to design a product insensitive to noise (low variance), so do this first and then bring it on target. He realized that having high accuracy with high variance (upper right figure) led to a low-quality product. If he were the archer's coach, he would first train them to reduce the variance and then work to get on target.

Taguchi also defined a loss function to describe how variance affects product quality. He referred to key performance indicators (KPIs) as product KPIs. Here is the logic of the Taguchi Loss Function: say the KPI is Target-is-Best, and its target is part of the specification. A threshold is also assumed symmetrical on each side of the target as in the left image in Fig. 2.

BACK TO TABLE OF CONTENTS

BACK TO TABLE OF CONTENTS

SECTION 9.0

Product Evaluation for X

Sections 7 and 8 focused on the best practices for evaluating the product design robustness and performance. Also necessary are the evaluations for cost, ease of manufacture and assembly, reliability, testability/maintainability, and environmental friendliness, all covered in this section. These evaluations have come to be known as design for cost (DFC), design for assembly (DFA), Design for Reliability (DFR), Design for Test (DFT), and Design for Sustainability (DFS), and so on, or generically—DFX. This is the TLA (three-letter acronym) section.

The DFX evaluations answer these questions:

- Is the cost of the product/components as low as reasonably possible?
- Are the components easy to manufacture?
- Is the product easy to assemble?
- Is the product reliable?
- Can the product's functionality be tested after it is in use and can it be maintained?
- Is the product environmentally friendly, designed with sustainability in mind?

BEST PRACTICE 9.1

Successful Product Development Organizations

Design for Cost.

BEST PRACTICE KEY CONCEPTS

- Cost requirements drive many products.
- The cost of machined parts is a function of their size, tolerances, and the number to be produced.
- Injection molded part cost is tied to the complexity of the mold.
- Regardless of the manufacturing process, the cost usually decreases with volume asymptotically approaching a fixed value.
- The cost of additive-manufactured components is generally not very dependent on the quantity.
- The cost of PCBs depends on the size, the number of layers, and the components.
- The cost of code depends on its function or the number of lines of code.

The Goal of This Best Practice

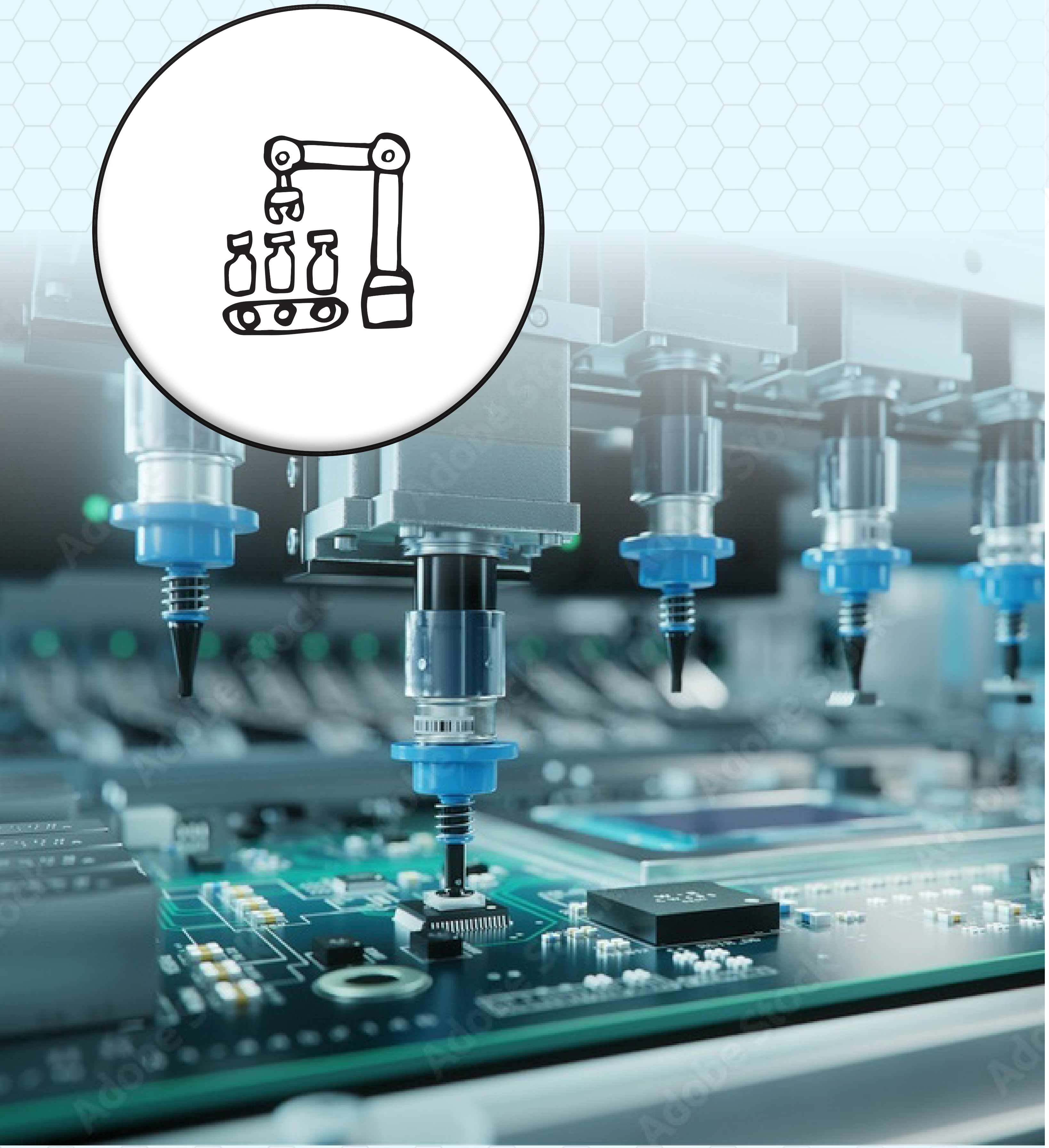
One of the most difficult and yet important tasks for a design engineer is estimating production costs. It is important to generate a cost estimate as early as possible in the design process and compare it with the cost requirements. A rough cost estimate is first generated during conceptual design or at the beginning of product generation. Then, the cost estimate is refined as the product is refined. Note that early cost estimates may be fairly accurate for redesign problems where changes are not extreme and the current costs are known.

As the design matures and the uncertainties go down, cost estimations converge on the final cost. This often requires price quotes from vendors and the aid of a cost estimation specialist. Many manufacturing companies have a purchasing or cost-estimating department responsible for estimating the cost of manufactured and purchased components. However, the engineer shares the responsibility, especially when there are many concepts or variations to consider and when the potential components are too abstract for others to cost estimate. Before describing cost-estimating methods for designers, it is important to understand what control the design engineer has over the product's manufacturing cost and selling price.

Since cost is usually a driving constraint, many companies use the term "Design For Cost" (DFC) to emphasize its importance.

While product cost is a combination of the costs to manufacture components or code and assembly costs, manufacturing

BACK TO TABLE OF CONTENTS



BEST PRACTICE 9.2

Successful Product Development Organizations

Design for Manufacture.

BEST PRACTICE KEY CONCEPTS

- Injection molded parts need thin walls and features that allow them to be easily removed from molds.
- Machined parts can get expensive as the features on them are increased.
- Printed components can have intricate features and are suitable for low-volume production.
- Printed circuit boards must be designed to support components and be easily mounted in physical structures.

The Goals of This Best Practice

Design For Manufacture (DFM) is widely used but poorly defined. Manufacturing engineers often use this term to include all or some of the best practices discussed in this book. Others limit the definition to include only design changes that facilitate manufacturing but do not alter the concept and functionality of the product. **DFM** establishes components' shape for efficient, high-quality manufacture. Notice that the subject of the definition is a component. DFM could be called DFCM, design for component manufacture, to differentiate it from Design For Assembly, DFA.

DFM's key concern is specifying the best manufacturing process for the component and ensuring that the component form supports the selected manufacturing process. For any component, many manufacturing processes can be used. For each manufacturing process, there are design guidelines that, if followed, result in consistent components and little waste.

Matching the component to the manufacturing process includes concern for tooling and fixturing. Components must be held for machining, released from molds, and moved between processes. The design of the component can affect all these manufacturing issues. Further, the tooling and fixturing design should be treated concurrently with the development of the component. The tooling and fixturing design follow the same process as the component design: establish requirements, develop concepts, and then the final product.

In the days of over-the-wall product design processes, design engineers would sometimes release drawings to manufacturing for difficult or impossible components. The concurrent engineering

BEST PRACTICE 9.3

Successful Product Development Organizations

Design for Assembly.

BEST PRACTICE KEY CONCEPTS

- Design for Assembly evaluation is only important if assembly cost is a significant part of the product cost.
- For hand, automatic, or robot assembly, thirteen DFA guidelines are used in industry to streamline assembly.
- DFA guidelines include part retrieval, handling, and mating.
- Estimating the time and cost of assembly is particularly challenging.

The Goals of This Best Practice

Design For Assembly (DFA) is the best practice for measuring how easy it is to assemble a product. Where Design for Manufacture (DFM) focuses on making the components, DFA is concerned with putting them together. Since virtually all products are assembled from many components and assembly takes time (i.e., costs money), there is a strong incentive to make products as easy to assemble as possible.

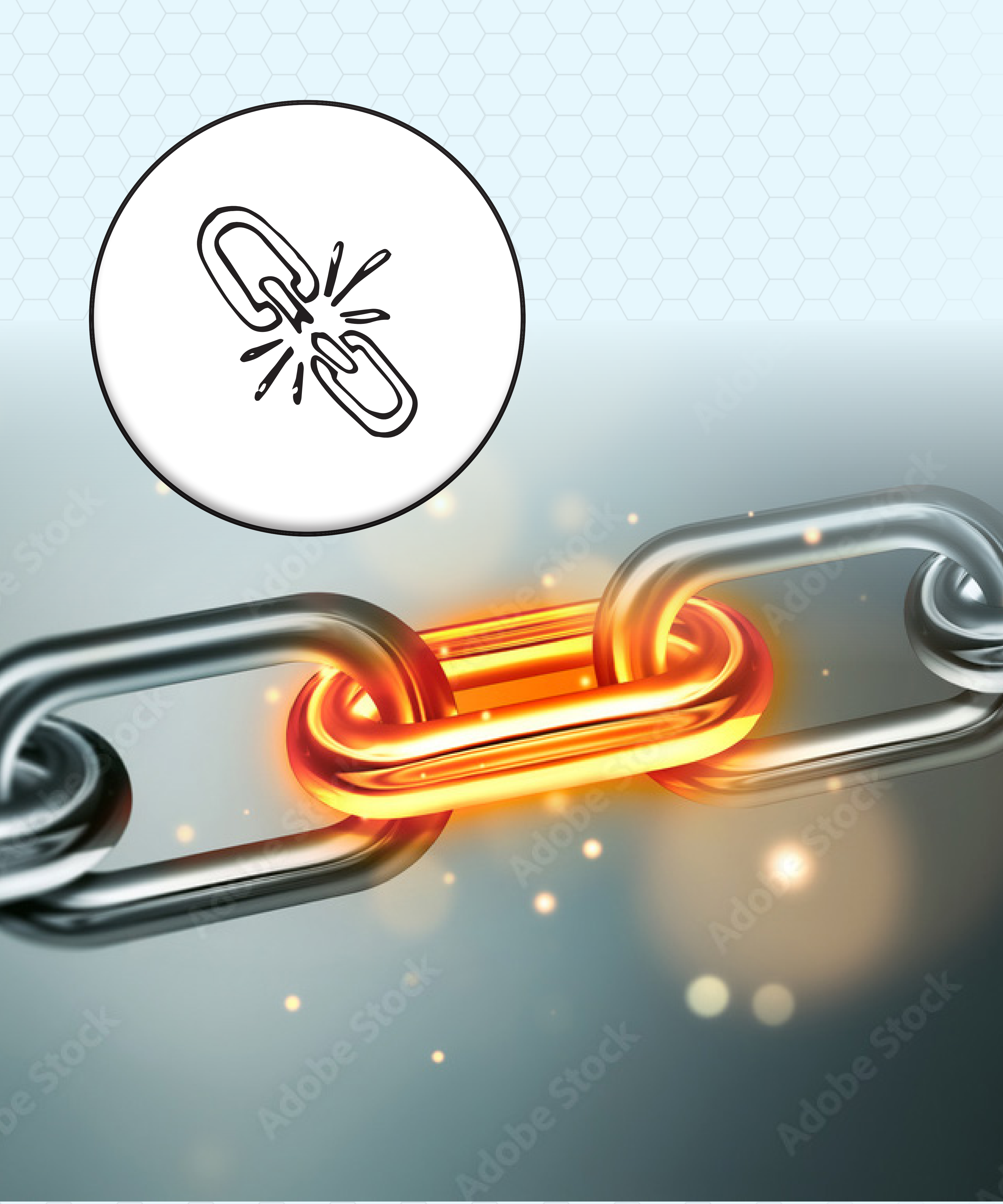
Throughout the 1980s, many methods evolved to measure design assembly efficiency. These methods require the design to be a refined product before they can be applied. The technique presented in this section is based on these methods and is organized around thirteen design-for-assembly guidelines, which form the basis for a worksheet (Fig. 1). Before discussing these thirteen guidelines, there are several important points about DFA.

Assembling a product means that a person, robot, or other machine must (1) retrieve components from storage, (2) orient them relative to each other, and (3) mate them. Thus, the ease of assembly is directly proportional to the number of components that must be retrieved, handled, and mated, and the ease with which they can be moved from storage to their final, assembled position. Each act of retrieving, handling, and mating a component or repositioning an assembly is an **assembly operation**.

Retrieval usually starts at some type of component feeder, ranging from a simple bin of loose bulk components to an automatic machine that feeds one component at a time in the proper orientation for a robot to handle.

Component handling is a critical consideration in the measure of assembly quality. Handling encompasses maneuvering the retrieved component so it is oriented for assembly. For a bolt to be

BACK TO TABLE OF CONTENTS



Successful Product Development Organizations

Design for Reliability.

BEST PRACTICE KEY CONCEPTS

- A reliable product has a low risk of failure.
- Failure renders a component, assembly, or system incapable of performing its intended function.
- Faulty Tree Analysis (FTA) is a method to discover failure events and their logical relationships.
- The Mean Time Between Failures (MTBF) is the average elapsed time between failures.
- Probabilistic Risk Assessment (PRA) is a quantitative method to analyze the risk of failure.
- There are four avenues for risk mitigation: avoidance, reduction, transfer, and acceptance.
- Failure Mode and Effects Analysis (FMEA) is widely used to understand potential failures.

The Goals of This Best Practice

Reliability is the probability that a product performs its intended function adequately for a specified period under specific operating conditions. A **failure** is unsatisfactory performance and can present a hazard if the consequence of its occurrence is sufficiently severe. When determining a product's reliability most organizations use **Failure Modes and Effects Analysis (FMEA)** for identifying failure potential**.** This Best Practice is useful as a design evaluation tool and aids in hazard assessment.

Before presenting FMEA, three other methods are discussed that provide material useful in developing it. All are widely used in product development organizations. First is **Fault Tree Analysis (FTA)**, a method to discover potential failure modes. The second, **Mean Time Between Failures (MTBF)** method, adds information about the expected time to failure.

Finally, **Probabilistic Risk Assessment (PRA)** is another method to identify failure modes and is an analytical method that uses MTBF to fully characterize the risk of failure.

This Best Practice is built around an example: the drive train of the Mars Exploration Rover

BEST PRACTICE 9.5

Successful Product Development Organizations

Design for Test and Maintainability.

BEST PRACTICE KEY CONCEPTS

- Maintainability is the ease of replacing parts or assemblies.
- The ability to test product function is a significant element of maintainability.
- Test and maintainability are important for expensive consumer products and all industrial products.
- In a throw-away society, there is little need for testing and maintaining many consumer products.

The Goal of This Best Practice

Design for Test and Maintainability (DFTM) addresses how a product is kept functioning while in use. The terms *maintainability, serviceability,* and *reparability* are often used interchangeably to describe the ease of diagnosing and repairing a product. **Maintainability** is the reaction to an event, as shown in Fig. 1. Three types of events trigger a change or potential change in how a product works, regardless of whether we are talking about hardware, electronics, or software. First, it can fail and require maintenance to get it back in working order. Second, maintenance may be undertaken to prevent failures. And third, there may be an upgrade to the product.

A significant part of a product's maintainability is how easily its functions can be tested for compliance with specifications. Often, the term **testability** is used for the ease of measuring the performance of critical functions. Testability is different from Test Driven Development, where testing is part of the design process used to evaluate how well the evolving product meets specifications. Here, tests diagnose failures that may occur in the final product during use. For example, specialized circuits for measuring critical functions are often included on the chip when designing integrated circuit boards. While some measurements are made during manufacturing to ensure that no errors are built into the chip, other measurements are made during use to diagnose failures, which is the concern here.

Testing may be directed at discovering what went wrong to cause a failure or as part of preventive maintenance, as shown in Fig 1. For example, all cars can be connected to an OBD (On-Board Diagnostics) code reader, shown in Fig. 2,

BEST PRACTICE 9.6

Successful Product Development Organizations

Design for Sustainability.

BEST PRACTICE KEY CONCEPTS

- The Earth does not have a pollution problem; it has a design problem.
- Most products end up in landfills.
- Measuring a product's greenness or sustainability is possible by developing a Design for Sustainability (DFS) score.

The Goals of this Best Practice

Design For Sustainability (DFS) is also called Design For Environment (DFE), green design, environmentally conscious design, lifecycle design, or design for recyclability. Treating environmental concerns as requirements in the design process began in the 1970s with the 1973 oil crisis. However, it was not until the 1990s that it became a prominent issue in the design community. Fig. 1 shows the major considerations of DFS. The arrows represent the flow of materials taken from the Earth or the biosphere and returned to it.

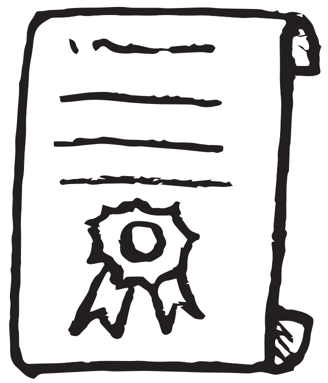
In the past, environmental issues related to products were primarily focused on the end of a product's life. When a product's useful life is over, one of three things happens to the materials used to manufacture and package it: they are disposed of, reused, or recycled. The latter part of the 20th century was mainly a disposal society, and little thought was given beyond disposal for most products.

By 2020, electrical and electronic equipment constituted around 4% of Europe's municipal waste and was increasing by between 16% and 28% every five years – three times as fast as the growth in other municipal wastes. Vehicles were also responsible for considerable waste, with 8-10 million cars, trucks, and vans disposed of yearly in the USA. In Europe, the number was around 14 million and growing, with the world figure around 30 million.

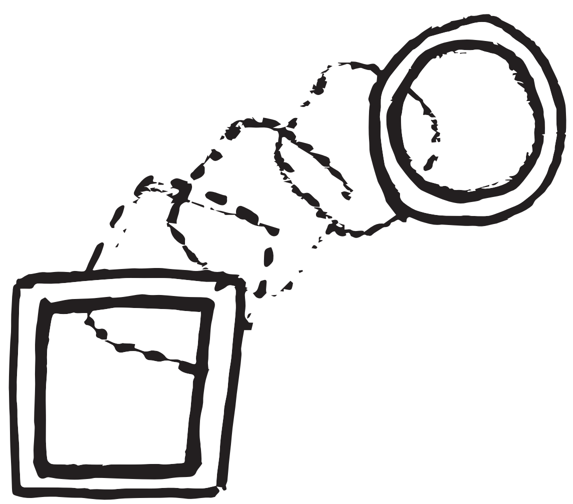
Whereas in the 1970s and 1980s, there was a design emphasis on disposable products, more industries now try to design in the ability to recycle or reuse parts of retired products. In 1995, 94% of cars and trucks scrapped in the United States were dismantled and shredded, and 75% of the content by weight was recycled. Currently, around 75% of the weight of a vehicle is already recycled; it is almost exclusively metals (i.e., steel, cast iron, and aluminum) that are readily identified and salvaged, together with components such as batteries.

By 2018 in the US, only 25% of glass and 9% of plastics were recycled. Virtually all lead-

BACK TO TABLE OF CONTENTS



BP 10.1
PRODUCT CHANGES



BP 10.2
INTELLECTUAL PROPERTY

BACK TO TABLE OF CONTENTS

SECTION 10.0

Post Design Considerations

These best practices answer the following design questions:

- How are changes to the product managed once it is in production?
- What intellectual property is important in product design?
- Why and how should ideas be patented?

This section addresses topics that, while not directly in the main flow of the design process, are very important for the designer's attention.

Best Practice 10.1 focuses on design activities that extend after the product has been released for production. Concern for post-design topics is vitally important during the design process. Prior best practices have paid close attention to this, but some specific topics still need to be addressed here.

Best Practice 10.2 describes how to protect the intellectual property that has been developed. It focuses on the patent application process.

BEST PRACTICE 10.1

Successful Product Development Organizations

Efficiently Manage Post-Release Engineering Changes.

BEST PRACTICE KEY CONCEPTS

- Product changes naturally occur during the design process.
- Product changes occur after product release.
- Product changes after release can be very costly.
- Ongoing product change can be part of a design philosophy.

The Goal of This Best Practice

Product changes occur throughout product development and beyond, as shown in Fig 1.

This figure shows the design changes versus time for two very different organizations. What is important here are three phases in the process. First is when the product

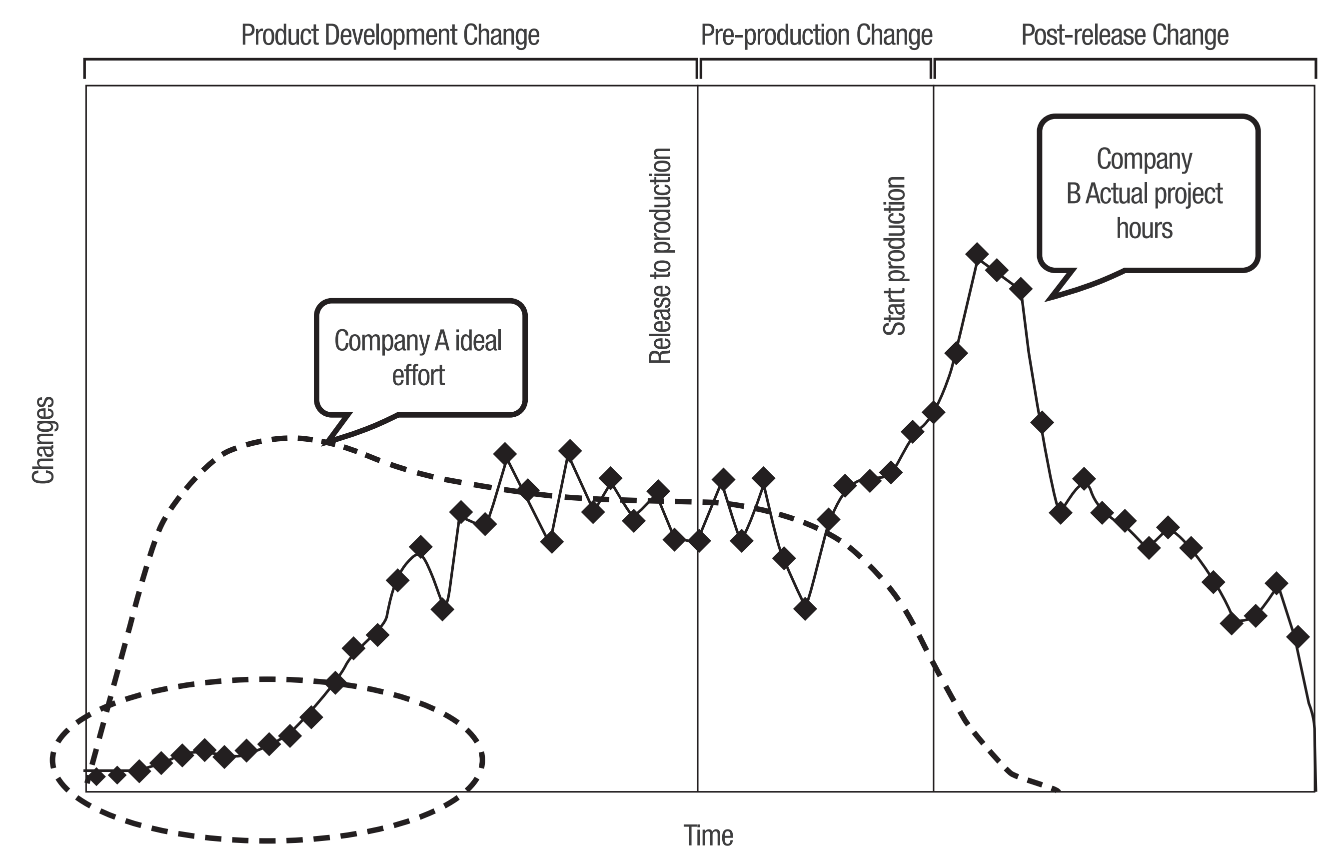
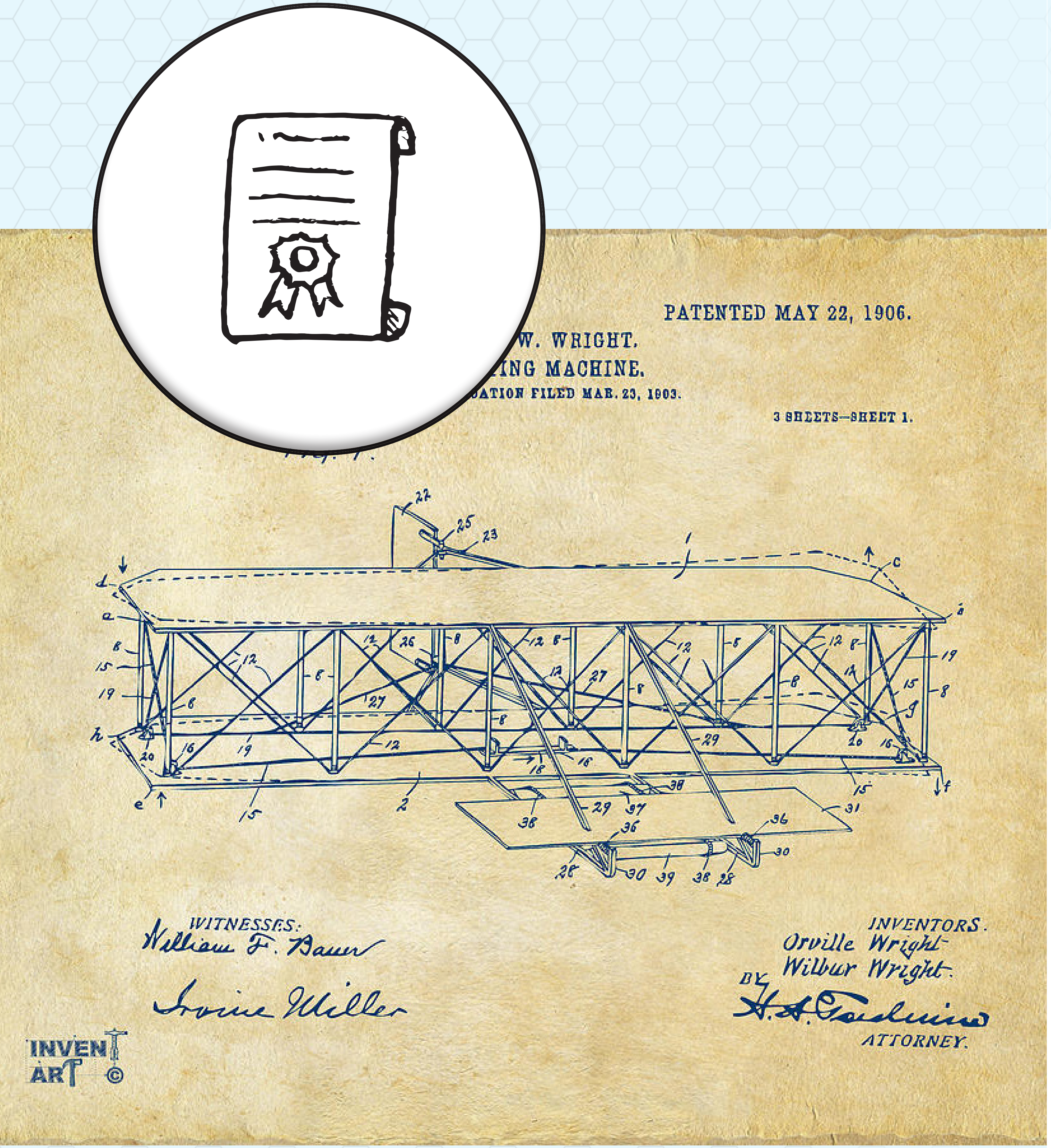


Figure 1. The changes made during the development of automobiles by two different companies.



BEST PRACTICE 10.2

Successful Product Development Organizations

Value and Manage Their Intellectual Property.

BEST PRACTICE KEY CONCEPTS

- All the ideas, analyses, and drawings are an organization's intellectual property.
- Organizations work hard to protect their intellectual property.
- Unique ideas can be protected through patents.
- Patents make ideas public.
- Patents reflect bets on unproved invention value.
- Patents give bragging rights and a license to litigate.
- The patent process is not difficult.

The Goal of This Best Practice

This best practice describes how successful organizations value their intellectual property. It outlines the patent process and, in doing so, provides details about the elements of intellectual property.

The Concept of Intellectual Property

All the design work done to develop a new product generates **intellectual property (IP)** or "trade secrets" owned by an organization. In the broadest terms, IP is the drawings, code, specifications, documents, prototypes, and so on generated during the design process. To protect this material, they may be able to apply for a patent, copyright, or trademark.

Successful product development organizations put great value on their intellectual property as it is an asset more important than their physical plant or employees.

Intellectual property laws are designed to encourage the creation of a wide variety of intellectual goods. To achieve this, the laws give individuals and organizations property rights to the information and intellectual goods they create for a limited period.

While details about patents are developed below, in the broadest sense, a **patent** is a government-granted monopoly to an individual or organization to build, sell, and use a concept and prevent others from doing so. Once granted, patents are available for all to read and leverage. Technological advancement is based on

BACK TO TABLE OF CONTENTS

and drawings attached.

Related Best Practices

Term	Definition	Best Practice
Intellectual Property	All the knowledge, drawings, documentation, test reports and other material generated during the design process that have value to the organization.	2.2

Resources

“Design patent application guide: A Guide To Filing A Design Patent Application” US Patent and Trademark Office, https://www.uspto.gov/patents/basics/apply/design-patent Accessed 2023.

“Google Patents”, https://support.google.com/faqs/answer/6390996 or https://patents.google.com/ Accessed 2023.

“Provisional Application for Patent: Provisional Patent Application Forms”, US Patent and Trademark Office, https://www.uspto.gov/patents/basics/apply/provisional-application Accessed 2023.

“Nonprovisional (Utility) Patent Application Filing Guide: A Guide to Filing a Utility Patent Application”, US Patent and Trademark Office, https://www.uspto.gov/patents/basics/apply/utility-patent Accessed 2023.

“Search for patents” US Patent and Trademark Office, https://www.uspto.gov/patents/search Accessed 2023.

GLOSSARY

Term	Definition	Defined in	Used in
A			
Accessibility (Product)	The product can be used by all regardless sex, handedness, disability or other stakeholder difference.	4.1	
Accuracy	A measure of how far from the truth or correctness a statement or value is. In design, the truth is often not known (uncertain) or knowable (aka lacks precision).	7.4	
Additive Manufacturing	The process of making three-dimensional solid plastic or metal objects from a computer model, aka 3D printing or rapid prototyping	1.8	
Additive tolerance Stack-up	The most common tolerance analysis by adding the maximum and minimum dimensions to estimate the stack-up or worst-case clearance or interference.	7.4	
Affinity diagram	Graphs diagrams for generating, organizing, and consolidating product or process information.	1.7	
Agile Design Process	A flexible system that sequences task based on decisions made during the project in a just-in-time manner.	5.3	3.1, 3.2, 4.6, 5.4, 5.5
Agile/Scrum	A 21st century design process developed for software but applicable, with modifications, to hardware and systems. Best applied when uncertainty and innovation are major factors. The method focuses on short iterations to learn quickly and to achieve a minimum viable product as rapidly as possible.	5.3	

Alpha testing	Testing by an internal team to support a research teams goals or tasks evaluating a new product or feature.	8.1	7.1
Alternative generation or Concept Generation	Developing multiple potential solutions for the product needs.	Section 6	2.1, 2.4, 7.1
Ambiguous	Items or information that are left to interpretation by individual team members. See VUCA+	2.3	
Analysis of Means (ANOM)	A DOE method to analyze experimental data based on analyzing the mean experimental value for each variable at each level.	8.3	
Analysis of Ranges (ANOR)	A DOE method used along with ANOM to analyze experimental data based on the range for each variable at each level.	8.4	
Analysis of Variance (ANOVA)	A statistical method for reducing experimental data based on analyzing the data variance. ANOM and ANOR are used in its place in this book.	8.4	
Assembly instructions	The steps for combining components into assemblies.	6.5	
Augmented and Virtual Reality Tools	Tools to support concept evaluation with virtual reality (VR), where the environment is totally artificial, or an augmented one (AR), where the artificial is mixed with the real.	1.6	8.1
B			
Backlog	A listing of tasks to be done or problems to be addressed. See Problem Backlog and Task Backlog.	5.1, 5.4, 5.3	10.1
Behavior	The resulting actions or performance of a product.	1.3	1.1

Benchmarking	A method to: (1) compare multiple products with respect to their performance against various targets and requirements; (2) to understand how a product works during product decomposition; (3) dissect (or reverse engineer) a product to understand how it is made (manufactured)	4.5	4.7, 9.1
Benjamin Franklin	An early user of Pro-Con Analysis	2.4	6.2
Best Practice	Professional methods that are accepted as being effective.	1.1	
Beta testing	External testing to support a research teams goals or tasks evaluating a new product or feature.	8.1	7.1
Black Swan	An unforeseeable event. See VUCA+.B44	2.3	7.1
BOM (Bill of Materials)	A parts list or index to a product.	6.5	
Brainstorming	One of the most commonly used idea-generation tools. The focus is to generate as many different ideas as possible.	6.1	
Brainwriting	An ideation method in which the collaborators write ideas in a shared space that all others can immediately read.	6.1	
C			
Champion	The owner of a concept or product who pushes it through its development.	2.4	
Change management	Controlling the documentation and manufacture of components and assemblies after they have been released for production.	10.1	1.4

Collaborative Sketching	A graphical idea generating method.	6.1	
Complex Information	When new information or a change in an unexpected way. Part of VUCA+	2.3	
Concept	An idea that is sufficiently developed to evaluate the physical principles or software structures that govern its behavior.	6.0	
Concept Generation (Alternative Generation)	Developing multiple potential solutions for the product needs.	Section 6	2.1, 5.1, 6.6, 7.1
Concurrent engineering	The simultaneous design of mechanical, electronic, and software elements of a product and their associated assembly and manufacturing processes	1.4	9.2
Configuration Design	The geometric problem of assembling components into a complete product (aka packaging design).	4.1	
Cost of Additive Manufacturing Components	A tool to estimate the cost of 3D Printing methods.	9.1	
Cost of Code	Methods to estimate the cost of writing code.	9.1	
Cost of Injection-Molded Components	A tool to estimate the cost of components made by injecting molten plastic into a mold.	9.1	
Cost of Printed Circuit Boards	A tool to estimate the cost of PCBs.	9.1	
Cost of Machined Components	A tool to estimate the cost of components made on a mill, lathe or other metal removing system.	9.1	
COTS (Commercial Off The Shelf)	Objects that can be purchased without the need for design effort.	6.3	

Creative solution	A solution to a problem that meets two criteria: it solves the problem in question, and it is original.	6.1	6.5
Critical Design Review (CDR)	A meeting to demonstrate that the technical effort is on track to complete the product and meets the requirements within the identified cost and schedule constraints.	5.5	
Customer	The person or organization that purchases (or may purchase) the product or services being designed. A type of stakeholder who may or may not be the end user.	4.2	
D			
Daily Standup	A short inspect-and-adapt meeting that allows the team to coordinate by sharing the previous day's accomplishments (and challenges) and individual plans for the coming day.	5.5	
Dark Horse ideas	A design approach that forces designers to explore "extreme" ideas that might not initially appear feasible.	6.2	
Decision by Chaos	Making decisions with no reasonable management.	2.4	
Decision by Coercion	An alternative's champion forces their favorite on the team.	2.4	
Decision by fiat.	Autocratically using authority to select their favorite alternative.	2.4	
Decision by inertia	Choosing the alternative most closely matched to what was done before.	2.4	
Decision by running out of time	Choosing an option just because time is up.	2.4	

Decision making	Choosing the best possible course of action by understanding the issue, developing measures to judge alternative solutions, developing alternatives, evaluating the alternatives and deciding what to do next.	2.4	3.1, 6.6, 7.2
Decision Matrix	An iterative evaluation method that tests the completeness and understanding of measures and alternatives, rapidly identifies the strongest alternatives, and helps foster new alternatives	2.4, 6.6	7.1
Decision Poker	A team method for making estimates.	5.2	8.2
Decomposition (or dissection)	Taking a product apart to find out how it is made, how it works and how it is manufactured.	4.5	2.1, 4.7, 6.6
Deliverable	A promised object, code, document, analysis result or other entity.	5.1	6.5
Design iteration	Repetition while developing acceptable form and function.	1.1	5.5
Design freedom	Since design is a series of decisions and each decision eliminates alternative possibilities, design freedom is lost as the process proceeds.	1.1	1.1
Design learning	Design is leaning about the evolving product.	2.1	1.1
Design of Experiments (DOE)	A method used to understand the effect of variables on the key performance indicators (KPIs).	8.3	8.1, 8.4
Design Phases	The part of the product life cycle that includes product definition, project planning, conceptual design, and product development.	1.2	8.4

Design Process	The managed structure of people's knowledge so they can make the best possible design decisions fulfilling a need with an object.	1.1	1.2, 8.1
Design recursion	A characteristic of design where the same design process is applied to the product, systems, subsystems, components, and features, with each interdependent on the others	1.1	1.1
Design Review	A meeting focused on reviewing prototypes, analytical results, drawings, charts or other representations of the evolving product.	5.5	2.1, 2.2, 1.1
Design space	The union of all possible products that meet the design requirements.	8.2	5.5
Design Structure Matrix	A diagramic tool to determine the sequence of tasks (for a project) or functions (for a system) and to cluster items into work or structure modules.	1.7	8.3, 8.4
Design Tools	A software or a physical object that either helps execute a design method or adds significantly to the design process.	1.6	1.3
Communication Support Tools	Communication tools include videoconferencing, and email.	1.6	
Planning Support Tools	Tools used for planning and project control.	1.6	
Lifecycle Support Tools	Tools to manage product information and the product itself throughout its life.	1.6	
Form Generation Tools	CAD, sketch capture and other tools to manage form generation.	1.6	
Performance Analysis Tools	Performance analysis tools help designers confirm the function of concepts and products.	1.6	

Human Interface Analysis Tools	These tools support the analysis of user-product interaction.	1.6	
Augmented and Virtual Reality Tools	Tools to support concept evaluation with virtual reality (VR), where the environment is totally artificial, or an augmented one (AR), where the artificial is mixed with the real.	1.6	
Information Relationship Tools	Information relationship tools such as flow charts, Data Flow Diagrams, Design Structure Matrices, Mind Maps, Affinity Diagrams, and UML help teams to develop and organize information	1.7	
Measurement Tools	Measurement tools allow data on physical measures such as heat, pressure, velocity, shape, waveform, voltage, current, or power.	1.6	
Design for Assembly (DFA)	Principles, guidelines, and analyses that focus on making the assembly process as efficient as possible.	9.3	3.2, 4.2, 6.5, 9.1, 9.2, 9.5, 9.6
Design for Additive Manufacture (DFAM)	A collection of specific design rules and tools that aid in the creation of components optimized for 3D printing.	9.2	9.1
Design for Cost (DFC)	Using cost estimation during the process to drive design decisions.	9.1	4.2,4.3
Design for Manufacture (DFM)	Principles, guidelines, and analyses that focus on how the product will be produced.	9.2	4.2, 4.3, 3.2, 9.1, 9.3
Design for Maintainability (DFM)	Principles, guidelines, and analyses that focus on how to improve a product to make it easier to maintain.	9.5	1.4, 4.2
Design for Reliability (DFR)	Principles, guidelines, and analyses that seek to make the product as reliable as is warranted.	9.4	1.4, 6.5, 7.3

Design for Sustainability (DFS)	Principles, guidelines, and analyses that seek to minimize the environmental impact of a product during production, use, and retirement).	9.6	1.2, 1.4, 7.3, 4.2, 4.3, 9.3, 9.5
Design for Test (DFT)	Principles, guidelines, and analyses that seek to make testing the final product to determine failures as easy as possible. Not to be confused with Test Driven Design	9.5	1.4, 4.3, 5.1
Design for Anything (DFX)	X = assembly, manufacture, cost, reliability, etc.	9.0	
Design Patent	A patent covering the look of an object.	10.2	
Design Structure Matrix (DSM)	A method that maps one engineering domain to another: e.g. problem to product, component to component, assembly to architecture). This is used to help identify clusters of connected systems, helping to establish a product's architecture.	1.7	
Direct Safety	Safety designed into the product.	4.2	
Dissection	See decomposition	4.5	
Digital Twin	A virtual representation of an object or system that spans its lifecycle. It is updated with real-time data and uses simulation, machine learning and reasoning to support decision-making.	1.8	
E			
Electronic Design Automation (EDA)	Tools that are used to design and verify integrated circuits (ICs), printed circuit boards (PCBs), and electronic systems in general.	1.6	
Embodiment design	Giving form to concepts.	6.5	6.4

End-of-life product phases	That part of the product's life cycle that includes retirement, disassembly, reuse, recycling and disposal.	1.2	
Engineering Change Notice (ECN)	A notice of an product modification made after the product is in production.	10.1	
Engineering specification	See "specification"	4.6	
Estimates	Best guesses for task time and cost.	5.2	6.6
Estimation (or agile) poker	A method that helps teams make estimates when information is uncertain.	5.2	
Evaluation	The act of comparing concepts to engineering requirements and optimizing performance	Sections 7, 8 and 9	2.1, 2.4, 6.5, 6.6
Expertise (types of)	Expertise is measured both in breath and depth. The agile community describes expertise as being of T, M, I or -- types.	3.2	3.3
F			
Factor of Safety	A factor included in the design of physical objects to account for uncertainty and lack of knowledge. Also see Safety Margin.	7.3	1.4, 6.5
Failure Mode and Effects Analysis (FMEA)	A method to for identifying and prioritizing potential failures and causes during product design (DFMEA) and manufacturing (MFMEA).	9.4	3.4, 5.3, 7.2, 9.5
Feature	The parametric, geometric, topologic or semantic focus of interest that can be treated as a single element that has a specific intent/purpose.	4.1	4.5

Feature Creep	When new functions or elements are added to a project that expanding the initial scope of the underlying problem and result in extending development time and costs	2.3	4.4
Fidelity	A measure of how well a model or simulation analysis represents the state and behavior of a real-world object.	2.1	5.1, 8.1
Fixed, stable interfaces	Designing fixed, stable interfaces allows for the independence of modules and better products.	6.3	
Flow chart	A diagramtic method to organize function, the flow of information, energy, control, materials, task work, and data.	1.7	
Focus Group	A meeting designed to find out what is wanted in a product that does not yet exist. It relies on the customers' imaginations.	4.4	
Fractional Factorial Experiments	Evaluations where carefully selected combinations of independent variables are tested.	8.3	8.1
Fault Tree Analysis (FTA)	A method to identify sub-system failures and how these accumulate to the system level for products being developed	9.4	
Full Factorial Experiments	Evaluations where every possible combinations of independent variables are tested.	8.3	
Function	The transformation of energy, material, signal, forces or information between, across, or within objects or the change of state of an object caused by one or more of these flows.	1.3, 6.4	1.3, 1.4, 1.5, 6.3, 6.2, 6.5 7.4, 8.1, 9.6

Function Model	Black box models that explain the functions associated with sub-systems and components. An important early step in concept development.	6.4	1.3, 1.7, 2.3, 4.2, 4.4, 6.1, 6.3, 6.5, 9.4, 9.6
G			
Galley method	A graphical idea generating method.	6.1	
Gantt Chart	A method to graphically represent known tasks, their order, schedule and progress .	5.3	5.4
Generate and Test	A weak approach to optimizing performance, exploring the design space one point at a time.	8.2	
Generative Design	AI methods to develop 3D structures that maximize the load-carrying capability with the lightest possible components	1.8	
Golden triangle	Representation of the balance between cost, time and quality	1.4	
Good enough for now (GEFN)	The balance (cost/time/people) needed to develop an object relative to the value of the function or form of the object at this point in development.	5.1	
H			
Hazard	A situation that, if not corrected, might result in death, injury, or illness to personnel or damage to, or loss of, equipment (What can go wrong?).	7.2	4.3
Hannover Principles	The basic principles of Design for Sustainability with respect to designing buildings and objects.	1.2	9.5, 9.6

Human factors	The users' interaction with a device as occupant of a workspace, as a power source, as a sensor, and as a controller	4.3	
I			
Incomplete information	Information about an object is missing. See VUCA+I	2.3	
Indirect Safety	Safety elements added to a product to indirectly protect people or other objects.	4.2	
Interface	The boundary between components, assemblies, modules, users, and other objects. Design starts at interfaces.	6.3	1.3, 1.5, 4.3, 5.3, 10.1
Intellectual Property	All the knowledge, drawings, documentation, test reports and other material generated during the design process that have value to the organization.	2.2, 10.2	6.5
Internet of Things (IOT)	In the IoT universe, individual products or systems within them have processors that collect and share information with others to make collective decisions about their operation.	1.8	1.5, 9.5
Introspection	Team reflection, debriefing+B188, and self-examination to discover how to improve their design process.	3.4	2.1, 3.1
ISO-9000	The International Standard Organization's quality management system that demonstrate that an organization has documented product development plan.	5.5	
Iteration	Iteration is revisiting a decision and re-executing tasks as a means of getting closer to an acceptable form or function.	5.5	6.5

J			
Journey Map	A block diagram of the steps taken by a product through its life cycle (a Product Journey Map) or a users' steps when interfacing with an object (a User Journey Map).	4.2	4.4
K			
Kanban	A visual workflow project management process that organizes tasks into small value-added increments.	5.3	5.4
Kanban board	An agile project management tool designed to help visualize work and manage work-in-progress	5.4	
Kano's model	A plot of customer satisfaction versus product function that helps in understanding how and why specifications exist and mature.	4.6	4.7
Key Performance Indicator (KPI)	A measure of product or system performance.	8.1	8.2, 8.3, 8.4
L			
Last possible moment	The latest time that a decision can be made without slowing down the project.	2.4	
Learning	See design learning	2.1	
The Lifecycle Assessment (LCA)	Measures a product's or process's greenness or sustainability throughout its entire life cycle.	9.6	
Linear Design Process	Linear processes consist of a series of pre-defined tasks	5.3	
M			
Maintainability	See design for maintainability	9.5	

Maturity (product)	A measure of how stable and known the technologies that are used ion a product.	7.1	4.1
Maturity (process)	A measure of how stable and known the tasks are to solve a problem	4.1	5.4,5.5
Mean Time Between Failures (MTBF)	Average elapsed time between product, system, or component failures.	9.4	
Mechatronics	The integration of mechanical, electrical, computational disciplines that result in design or products and manufacturing processes		
Meta information	Information about information, such as such as customer, author, approval date, revision, information type, project name, contact information, or page count.	2.2	
Method or Design Method	A set of steps and activities used to create or document information during the design process.	1.1	1.6
Mind map	A diagram for representing tasks, words, concepts, or items linked to and arranged around a central concept.	1.7	
Minimum viable product (MVP)	A solution to a problem that meets the minimum goals or targets, it is good enough for now. An MVP is often considered as "satisficing".		
Modality	The manner in which information is represented such as, textual, graphical, auditory, non-verbal gestures, and physical.	2.2	
Modeling	Building an analytical, virtual, or physical method to evaluate performance.	8.1	

Module+A253	A distinct system or assembly that provides unique functions and has interfaces designed so it can be treated as a single object added or removed from a larger system.	6.2	9.5
Morphological Charts	A table where each row is a characteristic or function that needs to be included in the concept, and the columns include various "means" to achieve each function. A complete concept is one means from each row combined with others.	6.1	
N			
Non-Disclosure Agreement (NDA)	A written agreement between organization to keep each other's IP secure	10.2	
Noise	Variation caused by parameters that are impossible to control or are chosen not to be controlled due to cost or other factors.	8.3	8.4
O			
Object	A system, sub-system, assembly, component, module, feature or unit of code to be designed.. Also often referred to as an artifact.		
Optimization	An iterative evaluation technique with the goal of maximizing performance. Formal Optimization is an analytical method that can only be used when there is a set of equations relating the variables to KPIs and VUCA+ effects are small or nonexistent.	8.2	8.1
Original Design	Developing a new process, assembly, or component	4.1	
P			

Pairwise Comparison	Comparing alternatives two-at-a-time to find the better alternative.	2.4	6.6
Parallel Design	See set-based design	6.2	
Parametric Design	Using algorithms to create complex, customized products or structures.	4.1	
Patching	The activity of changing or fixing a design without changing its level of abstraction. Patching does not add value to the effort and wastes resources (aka reworking). Contrast to refining.	6.5	
Patent	Patents can be obtained for an object or code that is 1) new, 2) useful, 3) non-obvious, and 4) realizable.	10.2	
Performance	The measure of function and behavior—how well the device does what it is designed to do.	1.3	
Personas	Wearing the hat of a specific stakeholder to aid in developing requirements.	4.2	

Plan	A detailed proposal for action that has the intent to achieve something through the completion of defined tasks.B128	5.3, 5.4	1.4, 1.6, 5.1
PMI method	Plus-minus-interesting method for idea generation similar to Brainstorming.	6.1	
Precision	Having low variation (see accuracy)	7.4	7.3
Preliminary Design Review (PDR)	Demonstrating that the design meets all requirements with acceptable risk and within the cost and schedule constraints	5.5	10.1
Probabilistic Risk Assessment (PRA)	A quantitative method to analyze the risk of failure.	9.4	

Problem Backlog	The issues that will need to be addressed to solve the design problem; the known requirements to meet, the tasks to do and other issues to resolve.	1.1	5.3
Problem Granularity	The scope of the problem, whether the object being designed is the entire product, a system within the product, a component, or a specific feature	4.1	
Problem solving style	Every member of the team has a personal problem-solving style. This "style" includes how they interact with others, manage information, deliberate, and reach conclusions.	3.2	
Problem Understanding	Refining the stakeholders needs into customers' requirements and engineering specifications.	Section 4	2.1
Problem Understanding (how much time to spend)	Engineers who spend around 7% of their time understanding and developing requirements and specifications have significantly better solutions than those who spend less.	Overview and Organization of Best Practices	
Pro-Con Analysis	A design support method based on listing the pros and cons for comparing alternative solutions	2.4	6.6
Product Decomposition	A method to tear-down (virtually or physically) an existing artifact to determine how it works (functions, behaves)	4.5	
Product	Any result of the design process, regardless of whether or not it is a one-off bookshelf, a space station, a consumer product, or a component of any of them.	1.1	
Product function	What an object does or should do, usually expressed by action verbs	1.3	

Product Life Cycle	The progression of a product from cradle to grave. There are four stages in the Product life cycle: Design, Production and Delivery, Use, End of Life.	1.2	
PLM (Product Life Cycle Management)	An advanced system to store and control product information throughout the entire lifecycle, from the design phase through its production and sales, its service requirements, to its ultimate retirement.	1.6	6.5
Product Life Stages	see Product Life Cycle	1.2	
Product Plan Review (PPR)	A review of the plan for realizing the product	5.5	
Product Portfolio Management	Approach to prioritizing and selecting backlogged projects.		
Product Proposal	The Product Proposal outlines the objective for the product as a beginning for the Problem Backlog.	4.1	
Product use phase	That part of the product life cycle that includes its use cleaning and maintenance.	1.2	
Products Liability	The special branch of law dealing with alleged personal injury or property or environmental damage resulting from a defect in a product.	7.2	
Production and Delivery Phases	That part of the product life cycle that includes product manufacture, assembly, distribution and installation.	1.2	
Program Evaluation and Review Technique (PERT)	A common method to make estimates based on optimistic, most likely and pessimistic assessments.	5.2	
Project Proposal	The identification of the work to be done to develop a product.	4.1	

Proof-of-concept prototype	A prototype (aka proof-of-function prototype) focused on developing the product's function for comparison with the customers' requirements or engineering specifications.	5.1	
Proof-of-process prototype	A prototype used to verify the manufacturing process.	5.1	
Proof-of-product prototype	A prototype developed to help refine the components and assemblies.	5.1	
Proof-of-production prototype	A prototype used to verify the entire production process.	5.1	
Prototype	A physical, analytical, graphical or other instantiation of a product or object, meant to: learn about object, test an evolving object, communicate with others, or resolve one or more issues during product development.	5.1	5.3, 5.4, 6.5, 9.1
Provisional Patent	A one-year placeholder for ideas that gives the inventor time to work toward perfecting the invention and exploring the market potential before the time and expense of applying for a utility patent.	10.2	
Pruning	Deciding which concepts to continue to develop. (aka down-selecting)	6.6	2.4
Q			
QFD (Quality Function Deployment)	A method to manage the development of requirements and specifications.	4.7	8.2
Quality	The level of satisfaction of a customer based on product performance	1.4	4.3, 7.4, 8.2
Quality Control	The inspection of incoming raw material and manufactured components for conformance to the design documentation.	7.4	
Quality measure	Any variable that is an indicator of product quality. These are generally identified when developing specifications and may be a measure of performance or any other significant product attribute.	8.2	
R			
Recursion	A characteristic of design where the same design process/method is applied to the product, systems, sub-systems, components and features with each interdependent on the others.		
Redesign	Making changes to an existing product.	4.1	
Refining	Making an object less abstract (or more concrete) through iteration. Contrast to patching.	6.5	
Reliability	A measure of how well product behavior is maintained over time	9.4	
Requirements or (Design Requirements)	What the object should do, a characteristic of, how it should support a user, or what it should be; can be constraints, criteria, wishes, demands, or goals. Requirements are refined into specifications.	4.3, 4.4	1.6, 1.7, 2.3, 2.4, 3.1, 4.1, 4.2, 4.4, 4.7
Requirements for Specifications	What is needed for a good set of specifications	4.6	4.1
Retrospective	A meeting to review the design process and develop improvements for it.	3.4	5.3
Reverse Engineering	The virtual or physical teardown of a product to learn how it is made/ manufactured. (see Benchmarking)	4.5	
Risk	An expected value, a probability that combines the likelihood of something happening times the consequences of it happening	7.2	4.3, 5.3, 6.5, 9.4

Robust Design	A method that focuses on developing products that function while being insensitive to noise	8.4	8.1, 8.3
S			
Safety	The desire to have low risk of causing injury or loss.	4.1	4.3, 6.5, 7.2, 7.3
Safety Margin	Another name for a factor of safety, commonly used in electronics.	7.3	
Scrum	An Agile project management method that adds more structure to Kanban with "sprints" that define short development cycles.	5.3	
Selection Design	A method that involves choosing one item (or maybe more) from a list of similar items	4.1	
Sensitivity analysis	A method for evaluating the statistical relationship of parameters and their tolerances in a design problem to support trade-offs.	8.2	7.4, 8.1, 8.3
Set-based design	A design philosophy to explore multiple sub-solutions and alternatives, carrying them through modeling, prototyping, and detailing them to delay decision-making (aka parallel design).	6.2	6.1, 6.3, 6.4
Simplicity	Striving for the ideal of providing the needed function with the fewest components and assemblies.	6.5	
Specification or engineering specification	A formalized requirement includes a subject, units, target, and thresholds	4.6	1.3, 2.3, 2.4, 3.1, 4.3, 4.7, 6.6
Spiral learning	Learning structured as spiral where each topic is built on the previous work on the same topic.	2.1	

Sprint	Short development design work based on 2-4 week cycles of: Organize, Plan, Do, and Review.	5.3	
Sprint Review	A meeting held at the end of the sprint review portion to critique the delivered sprint solution.	5.5	
Stage-gate	A sequential design process for mature products and systems with scheduled review/approval meetings	5.3	5.4
Stakeholder	A person or organization that interacts (directly or indirectly) with the product being designed/ manufactured. An alternative definition is everyone downstream of the designers who comes in contact with or is affected by the product. Stakeholders are the primary source for requirements. (Customers and users are types of stakeholders)	4.2	4.7
Statistical Tolerance stack-up	An accurate method to estimate the gap/interference.	7.4	
Stories	A method used in developing product requirements. A story has a specific format ("as a <user>, I would like <function, behavior> because <justification>")	4.4	
Sub-system	A system within a larger system		
SWOT Analysis	A decision making method based on identifying Strengths, Weaknesses, Opportunities, and Threats		
System	A group of interacting physical, virtual, or integrated objects performing a specific function.	1.3	

System Definition Review (SDR)	A meeting that occurs at the end of conceptual design and is used to examine the proposed system architecture and the functional elements that define the concept	5.5	
Systems Specifications Review (SSR)	A meeting that occurs at the end of the product definition phase. It ensures that the functional and performance requirements defined for the system will satisfy the product need.	5.5	1.7
T			
Target	The desired, ideal level of performance aimed for. (Also see Threshold)	4.6	4.7
Task	A unit of work with clear deliverables	5.1	3.1, 4.1, 5.3
Task Backlog	A listing of the tasks that need to be done and the order to do them.	5.3	1.7, 3.1, 4.1, 5.1, 5.4
Task Board	An Agile project management tool designed to help visualize work and manage work-in-progress (aka Kanban Board).	5.4	5.1
Tack Criticality	Deciding which tasks to work on is a function of relative criticality, a function of each task's; Importance, Technical difficulty, Dependency, Uncertainty, and Lead Time.	5.3	
Team building activities	Activities that can be used to build strong teams	3.1	
Team characteristics	The ten characteristics of successful teams.	3.1	
Team contract	A working agreement among team members that state team member's roles, goals, and performance expectations.	3.1	
Team Environment	Successful product development organizations ensure that the physical, virtual and organizational environment is conducive to good design and design team support.	3.3	
Team Health	Ensure that the team operates efficiently by monitoring and acting on the symptoms/causes/remedies for team performance issues.	3.4	3.1, 3.2, 3.3
Team meeting minutes	A record of what was discussed, what decisions were made and commitments for future work.	3.1	
Team member roles	A team member's responsibility on the team.	3.2	3.1
Team of teams	The organization structure where there are identified teams within teams.	3.1	
Team structure	Teams are organized in one of three patterns, centralized (i.e. job shop), system (dedicated to a project) or hybrid (a mix with a discipline (job shop) and a project manager).	3.2	
Technology Readiness Assessment	A measure of a technology's maturity and readiness to be used in a product.	7.1	6.6, 7.2, 8.1
Test Driven Development (TDD)	A method that emphasizes that tasks have measures, targets and thresholds and the tests that prove the task is done.	5.1	1.6, 8.1, 9.5+D70
Threshold	A design specification threshold is the level of performance minimally accepted. (Also see Target)	4.6	4.7

Tolerance analysis	Assessing how variations in manufacture influence (1) the performance of a product or (2) the ability to assemble it.	7.4	8.4
Tolerance Stack-up	Assessing how assemblies fit or interfere. See Worst case or Statistical stack-up	7.4	
Tool	See "Design Tool".	1.6	
Trade-off	Decision making process that includes accepting a lower value/ performance in one measure in exchange for a higher/value performance in a another.	8.2	4.6
U			
Uncertain	Information is not precise, its description or value may vary. See VUCA and accuracy	2.3	
User	A stakeholder who makes use of the product.	4.2	
User Centered design	A popular term for understanding a user's demands, priorities, and experiences.	4.4	
User Experience (UX)	Concern for the entire users' experience with the object	4.3	
User Interface (UI)	Concern for the interface between the user and object being designed.	4.3	
USPTO	The US Patent and Trademark Office	10.2	
Utility patent	“Utility” is effectively synonymous with “function,” so the claims in a utility patent are about how an idea operates or is used.	10.2	
V			
Value	The monetary worth of a function or performance level		

Variant Design	A variant is a customized product designed to meet the needs of a specific customer.	4.1	
Variation	The distribution around a mean value.	7.4	8.4
Voice of the customer	A commonly used term to the requirments emphasizing the importance of the stakeholders' needs.	4.2	4.1, 4.7
Volatile	Information is changing and evolving (mean value moving) - See VUCA+		
VUCA	The dimensions of uncertain information: Volatile, Uncertain, Complex, and Ambiguous.	2.3	
VUCA+	The dimensions of uncertain information: Volatile, Uncertain, Complex, and Ambiguous + Incomplete + Black Swans	2.3	2.4, 6.6, 7.2, 8.1, 8.4
W			
Warnings	The weakest form of safety added to a product.	4.2	
Waterfall	see Stage-gate		
What to do next?	The primary question during design.	2.4	8.1
Worst Case Tolerance stack-up	The most common tolerance analysis form is adding the maximum and minimum dimensions to estimate the stack-up or worst-case.	7.4	
Numerical			
5 whys	A method to get to the root of an issue.	3.4	6.2, 10.1
6-3-5 Method	6-3-5 is a method is typically used in conceptual design for expanding and exploring the design space. It is like brainwriting with some additional structure.	6.1	