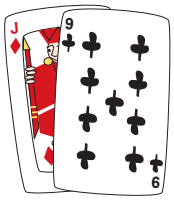
7. Create the solution - The Solution Domain of Information

The optimal solution is always the result of an iterative process in which the initial concept is first created and then gradually refined to perfection. Refining the initial concept requires access to more detailed information about the customer, needs, functions, and processes. Therefore, you need to revisit the other information domains several times during concept development. This iterative approach to filling the domains with value-critical information increases your chances of creating a WoW product.

Creating innovative solutions to value-increasing challenges is one of the areas where it's hardest to achieve improvement. Creativity is myth-shrouded, and many so-called "truths" are incorrect. I often compare it to playing blackjack.

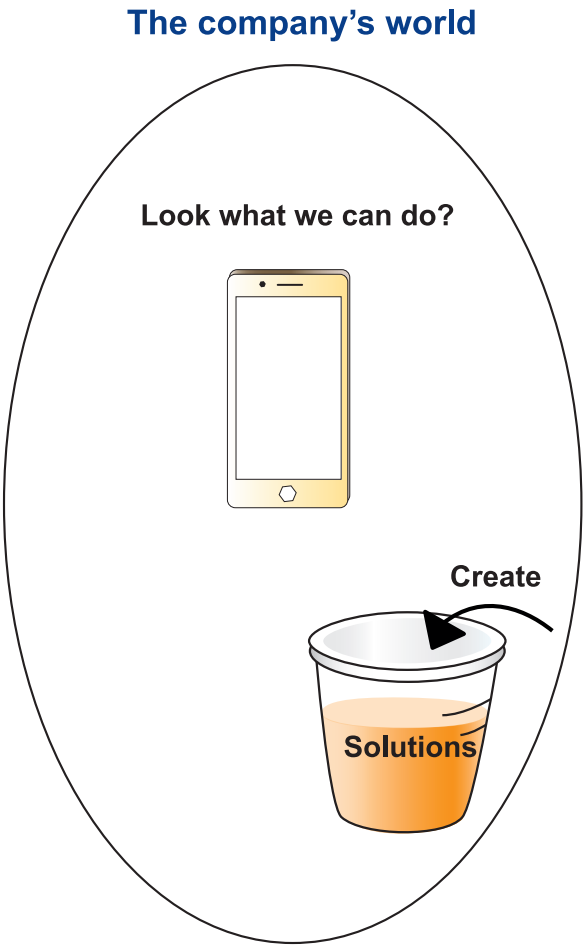


You can play blackjack, have luck on your side, and win. Most people who play loose. Unfortunately, that's the level at which some product development is done today.

Card counting is an advanced playing technique for blackjack. A skilled card counter gains a long-term advantage over the casino and wins in the long run. The method works because it increases the probability of winning. However, it requires both knowledge and skill from the player. It's not as simple as just sitting down at the table and taking a chance. The method is so successful that all casinos use different methods to detect card counters, and when they are discovered, they are thrown out and banned from the casino.

For product development, I have developed a method comparable to card counting in blackjack. However, the method requires more knowledge and skill than just taking a chance. The companies that apply this method increase the probability of success. They gain a long-term advantage over their competitors.

However, taking a shortcut is much easier and more comfortable, and sometimes it works. That's why worthless methods like wild brainstorming and thinking outside the box are still practiced.



This chapter is, therefore, not for the average development company or engineer. It's only interesting for those who want to learn a method that provides long-term wins, a method that requires more knowledge and skill. But if you learn to master this chapter, you'll become a brilliant development engineer, and on your resume, you'll also probably have WoW products.

This chapter contains the following:

7.1 Six systematic tactics for value-driven concept development

The best starting point to create a WoW product is your functional model of the benchmark and the list of value-increasing challenges created in the Functional domain. It contains all your options for increasing customer value. But before I start explaining more powerful tools than wild brainstorming, you need to understand functional modeling tactics. Functional modeling tactics are for product development, what card counting is for blackjack. However, functional modeling tactics are free for anyone willing to learn and use them.

Due to the strategic importance of functional modeling, it is essential to understand the different options available first. Later, I will discuss in what sequence the tactics can be used or when each tactic is favorable to use. These tactics help you accelerate the customer value development of your product. First, a quick introduction of the tactics:

- **Re-use:** Identify and retain the functional providers contributing the most value to your product or benchmark. These functional providers will remain unchanged in the new product.

- **Refine:** Optimize, trim, increase robustness, and/or reduce the costs of nearly perfect functional providers. This process enhances their value, making them suitable for your new product, and candidates to be re-used in future concepts.

- **Re-duce:** Systematically remove functional providers with low-value contributions, thereby reducing the complexity and cost of your new product. The cheapest component is always the one you have removed.

- **Re-inforce:** Introduce new functional providers that increase the visible value to customers. This process enhances the concept's performance, functionality, or appeal.

- **Re-form:** Modify or transform how the Main function is carried out in your benchmark. If done right, it can lead to a significant improvement in performance or elimination of unwanted functions. Many WoW products are born using this tactic.

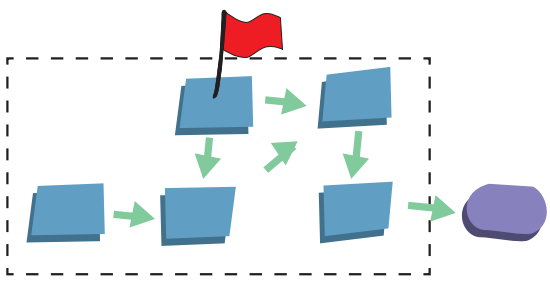
- **Replace:** Eventually, all products will be wiped out by new products with higher customer value. How can we survive the shift to the next S-curve?

The four first tactics are primarily aimed at accelerating value development for a product along an existing S-curve. The last two are based on the fact that no S-curve lives forever. All companies must also continuously test and challenge the existing S-curve. A more detailed discussion regarding the tactics is below.

7.2 The Re-Use tactic

The myth that creating a WoW product requires starting from scratch on a blank sheet of paper and thinking outside the box is just that - a myth. The idea that you need to throw out everything old and start anew is more research than product development. Some of your value-increasing challenges may indeed trigger the initiation of research projects. However, a new technology can take years or even decades to master, scale up, and fully implement in production. Consider yourself fortunate if you are lucky enough to have someone else's research ready to go.

For the rest of us who are not that fortunate, a more practical approach is to identify existing elements that can be re-used and carried over in the new product. Put a virtual "red flag" on subsystems, components, or process steps that work well and shouldn't be touched. The more red flags you use, the smaller the project scope, the lower the risks, and the fewer resources and time required to complete it.



It is also easy to forget about the learning curve. The power of the learning curve is that each time something is repeated, it becomes quicker, has better quality, and requires less effort. Every time you change something, your learning curve starts over. A company that constantly changes everything will never fully utilize the power of the learning curve. The beauty of the learning curve is that all you have to do to make it work for you is put out virtual red flags. It's probably the simplest way to cut costs.

There is an internal re-use loop and an external re-use loop. The internal loop is about re-using existing elements in your existing products. The external loop moves selected subsystems or components to your supply chain. Elements that are not critical, and can be produced elsewhere at lower cost and higher quality. What you should make in-house and what you shall buy from outside is part of the re-use tactic.

Unfortunately, some developers take pride in creating entirely new solutions from scratch, even when it is unnecessary. While still on the same S-curve, companies or development teams boast about the radical nature of their new product. They proudly proclaim that *"not a single component of the old product remains."* In my view, this is not something to brag about, but rather a symptom of a toxic macho attitude, where personal shine takes precedence over what is best for the company.

7.3 The Refine tactic

This tactic increases customer value by refining nearly perfect functional providers, such as subsystems, components, or processes. Elements that can be fixed and used in your new product and be candidates for re-use in future products.

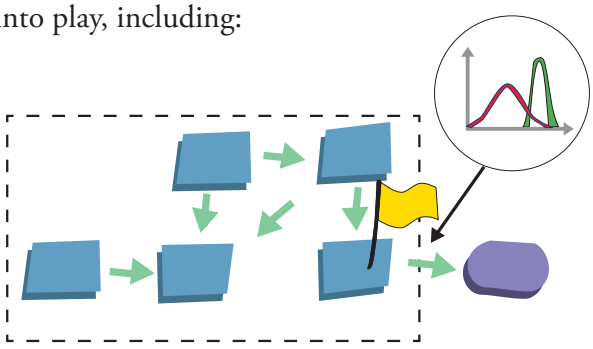
Here are some reasons why you may need to refine a functional provider in your concept:

- Poor design: failure to convey the intended identity or company image or suffering from poor ergonomic design.
- Over dimensioning: excessive margins due to a lack of knowledge or tools to optimize.
- Tight tolerances: tight tolerances should be used only for critical areas directly impacting customer value. Tight tolerances are expensive.
- Conflicting requirements: for example optimized from a manufacturing perspective but overlooking how it will affect the customer.
- Variability: not all customers receive the same level of quality.
- Robustness: failing to withstand natural variations in usage by the customers.

Sometimes, it is a kind of hygiene level - a set of issues that must be cleaned up, preventing the customer from experiencing the WoW factor. You might have a product that looks great on paper, a genuine WoW product, but it does not function entirely as intended when it meets reality. This is where the re-fine tactic comes in, to identify and eliminate those pesky problems that prevent the product from living up to its full potential.

It's about creating a solid foundation, a baseline of quality and functionality that allows the product to shine and deliver the desired customer experience. By addressing these hygiene factors, you can ensure that your product is not only innovative but also reliable, efficient, and effective. Ultimately creating a WoW experience for your customers. The re-fine tactic targets various design issues or problems and costs in manufacturing, such as scrap and rework, as well as costs arising at customer locations, including repairs and breakdowns.

When implementing the re-fine tactic, yellow flags are added to the functional model, signaling areas that require refinement rather than a complete redesign. This distinction is crucial, as it requires using different tools and techniques.
Re-fine focuses on optimizing and perfecting existing subsystems, components, or processes. To achieve this, a variety of specialized tools come into play, including:



- Industrial Design, which enhances the product's aesthetics and user experience.
- Design of Experiments and Robust Design, which help optimize performance and reduce variability.
- Computer Simulations, which enable virtual testing and analysis.
- Finite Element Method, which provides detailed insights into structural behavior.
- Design for X (DFX), is a set of methodologies that optimize products for specific characteristics, such as manufacturability, assembly, or maintenance.

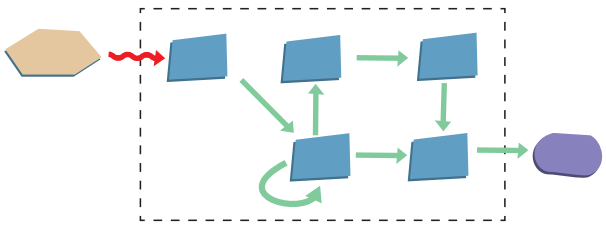
By using these and other tools, teams can refine and perfect existing designs, eliminating inefficiencies and improving overall performance without designing something new. The re-fine tactic is crucial to any continuous improvement strategy, such as Six Sigma, Kaizen, Quality Circles, or Total Quality Management.

7.4 The re-duce tactic

I often jokingly say that the most cost-effective and highest-quality component in a product is the one that has been removed. At first glance, the statement may seem absurd.

I have met many product development engineers looking at me like I am a complete idiot after arguing that an expensive or complicated subsystem needs to be removed. But the product will not work without it! They exclaim. Of course not, I say, not if the product remains the same as it is today.

In reality, many functional providers are old quick fixes for problems encountered in the past. For example, the oil got hot, so a cooling flange was added. That made it better, but not good enough, so a fan was added to the cooling flange. The fan



was noisy, and the customers complained, so a noise hood was added. I think you get the picture. Most products have evolved in this way through a series of quick fixes. We call the principle "Adding Waltzes". An old, worn-out refrain that many companies know all too well.

The quick fixes might have been the right choice at the time. My point is that these quick fixes can lead to complicated and expensive subsystems that may not be necessary, and a fresh look at the design can lead to simpler and more effective solutions. However, someone needs to break the spell.

By eliminating unnecessary components and subsystems, teams can significantly reduce product costs without compromising performance or customer value. However, it's essential to ensure that the removal of a component doesn't jeopardize the product's functions or overall performance.

New technologies and materials are continually being developed, offering opportunities for better and less expensive solutions. By embracing the reduce tactic and regularly reviewing and trimming products, teams can ensure that their products remain competitive, efficient, and effective over time.

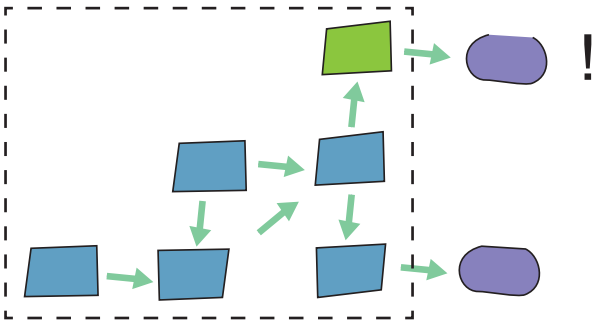
The problem-solving algorithm in Triz, which will be discussed later in this chapter, contains tools for accomplishing this tactic.

7. 5 The re-inforce tactic

The re-inforce tactic focuses on complementing the functional model with additional providers, such as new subsystems, components, or process steps. It is by far the most popular tactic used by development engineers. Two factors are driving this tactic:

- The quick-fix solution, a simple way to solve problems if time and resources are lacking.
- The desire to stay competitive by adding new features in the hope of improving customer value.

However, this tactic, while popular, can lead to



unintended consequences. The obvious and major drawback is that additions always add cost and complexity to your product. Cost and complexity, which may not be justified in the eyes of the customer. As products become more complex, they can also become more challenging to use and understand, potentially alienating the customers they aim to satisfy. The principle is right, but the way to make it happen is wrong.

An example of the above could be as follows. The company develops a way for the user to set a parameter to improve the product's performance. Improved performance increases customer value. The process for the user to learn, understand, and set the parameter reduces customer value. The intention is good, but the result might be bad. The increase in performance may not compensate for the operator's inconvenience in setting the parameter.

For the re-inforce tactic to be successful, it is imperative that the customer perceives the added value. Simply adding features is not enough. Those features must be meaningful and enhance the overall customer experience. The risk of failure is notably lower when the addition is associated with a new function that fulfills a previously unmet need or targets gaps in customer satisfaction.

Given its prevalence in many companies, I am confident you are already familiar with the re-inforce tactic. However, I hope this discussion will make it easier for you to recognize its use within your development projects. When you observe this tactic in action, because it will inevitably occur, let it serve as a warning signal. It is essential to evaluate whether the additions being made genuinely enhance customer value or are merely an attempt to respond to immediate pressures. If the situation calls for a necessary quick fix, document it. This will help ensure the issue can be revisited and addressed more thoughtfully in future projects.

In summary, while the re-inforce tactic can provide opportunities for innovation and increased customer value, it must be used judiciously.

Organizations should strive to balance adding new features and maintaining a clear focus on customer value. By understanding the implications of this tactic and recognizing its potential pitfalls, you can make more informed decisions.

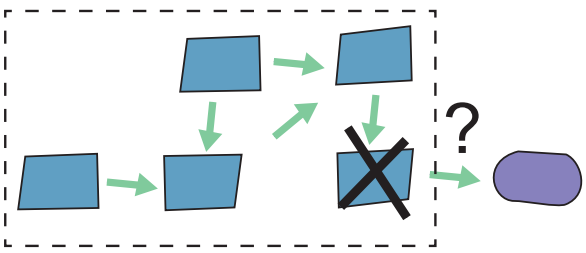
7.6 The re-form tactic

The re-form tactic is a powerful approach that has given birth to many innovative WoW products. This tactic involves focusing on the functional providers, such as subsystems or components, that perform the product's Main functions. The goal is to radically change these functional providers, thereby achieving a quantum leap in customer value. This is achieved by substantially increasing the performance of the Main function, adding a new important Additional function, or eliminating a serious Unwanted function.

Unlike incremental improvement tactics, the re-form tactic aims to bring about transformative changes that can propel the product into new market segments or position it at the beginning of a new S-curve. Correctly identifying the main functions of the product and the functional providers that perform them is a crucial step in the re-form tactic. The re-form tactic is often more complex and challenging to implement than previous tactics, as it involves changing the product's core.

A few avenues for you to explore can be:

- Solving feature conflicts instead of compromising can lead to ingenious solutions. Often, improving a product is hindered by conflicting features. Instead of finding a compromise, try to eliminate the conflict altogether. Three options to find a creep-hole around such a conflict are separating the conflicting features in time, space, or structure. For instance, a componen-



can be made strong and lightweight by being thick in one area, thin in another, or thick at one moment and thin in another. The Xylem wastewater pump (the N-pump) is a great example of solving a feature conflict using time and space separation principles.

- Transforming a rigid provider (subsystem or component) into a flexible one is also a classic example of the re-form tactic. Products that can adapt to varying conditions always have higher customer value. The adjustable wrench is a classic example of this, where making one half of the wrench flexible in relation to the other half enabled it to be used for nuts of different dimensions.

- Using a different technology to perform a function can make the impossible possible. What seems difficult or impossible with one technology can be easily achieved with another. The development of airbags, for instance, was hindered by the challenge of inflating them quickly. However, using pyrotechnics made it easy, as large gas volumes can be generated through explosions. A mechanic tends to think of mechanical solutions, an electrician in electrical solutions, and a chemist in chemical solutions. It highlights the importance of considering different technological solutions and collaborating with experts from various fields to find innovative solutions.

By mastering the powerful tools in Triz, your chances of successfully using the re-form tactic will dramatically increase. Some of the Triz tools will be introduced and discussed later in this chapter.

Ultimately, the re-form tactic requires a culture of curiosity, embracing a mindset open to new ideas, willingness to take risks, and commitment to continuous learning and experimentation. By adopting this approach, organizations can unlock the full potential of the re-form tactic and create innovative products that exceed customer expectations and drive business success.

7.7 The re-place tactic

Your product and business are constantly challenged from two directions:

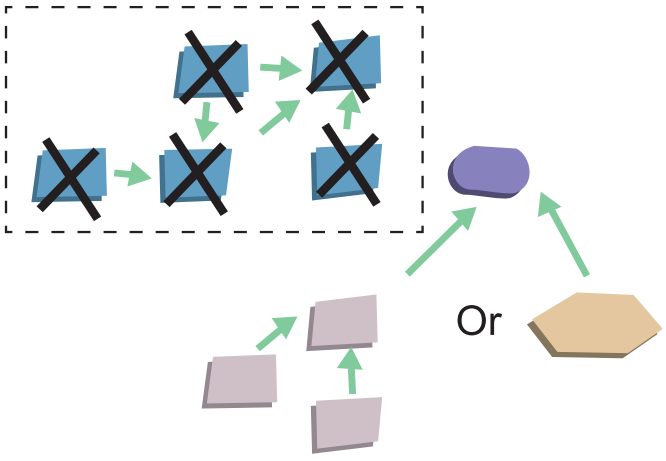
- The supersystem at a higher hierarchical levels.
- Competing solutions and competitors at the same hierarchical level.

All products are, in one way or another, part of a supersystem that also constantly develops toward higher customer value. The function performed by a watch is also performed by the mobile phone, the computer, the car, and so on. Today, you no longer need a separate watch to tell you the time. You may keep it anyway because it provides status, but in that case, its Main function has changed.

Studying the development of different products over longer periods shows that the supersystem is continually improved so that underlying subsystems can be made less expensive, simplified, or eliminated. This applies especially to subsystems or components that perform Support functions or reduce Unwanted functions within the supersystem.

The re-place tactic must be kept active within all companies. There's always someone higher up sawing on the branch your company is sitting on. Your only consolation and the best means of survival is taking over other products' functions from lower hierarchical levels.

To re-place or to be replaced is an eternal life-and-death struggle for all products.



Your biggest competitor may be the supersystem. A breakthrough at the supersystem level wipes out everybody, your product, and all your competitors.

You also have competitors at the same hierarchical level. Competitors outperforming your company on the re-form tactic may make your product obsolete. Almost all mechanical watch manufacturers were wiped out when the quartz movement watch was introduced. The mechanical watch went from a mainstream product to a niche product. Unfortunately, many companies have fallen victim to new technologies or been outmaneuvered by competitors who have mastered the six tactics better than their competitors.

"If our current product becomes a relic, it will only be because we have outdone ourselves with something even better."

That is the only strategy to save your company from eventually being wiped out by competitors or changes in the supersystem. You have to build internal competition into your company's DNA. Compartmentalize and protect radical new ideas from the rest of the company. Give them room to grow, become strong, and challenge the core of your business. The biggest competitor to your current product should preferably be some other part of your company.

7.8 How to apply the different tactics

The six tactics are the equivalent of card counting in blackjack - when applied correctly, they significantly increase the probability of success. If you are lucky, using one tactic could be enough to create a WoW product. However, if you constantly use several in every project, your probability of success will increase. Eventually, you will outperform the lucky ones.

Not all these tactics need to be applied to the functional model of your product or benchmark. In practice, you probably lack the time and resources to apply them all. I recommend that you always use a minimum of three in all development projects. You need to apply them in a sequence that maximizes the value increase of your product. Every project and situation is unique, so you must find your optimal sequence. But most probably, it will start with the re-use tactic and end with the re-duce tactic. However, all tactics should all be carried out with a certain degree of regularity and remain an active process within your organization.

A company that masters these tactics will outperform its competitors in the long run. The six tactics provide a high-level perspective, revealing general patterns and allowing you to rise above the noise of technical details and minor tweaks, value-critical information that should form the foundation of your decision-making in the solutions domain.

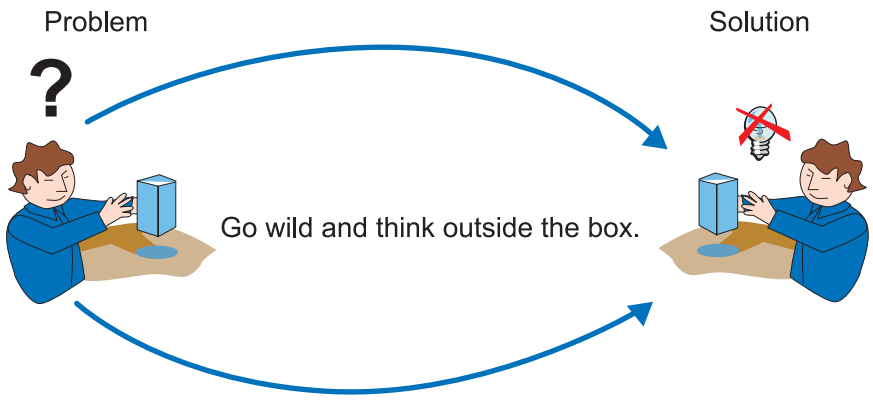
The first step could be to evaluate how proficient your company is in the various tactics. Are they being used frequently? How well do you apply them? How many are typically used in a development project? Rate your performance on a scale of excellence on:

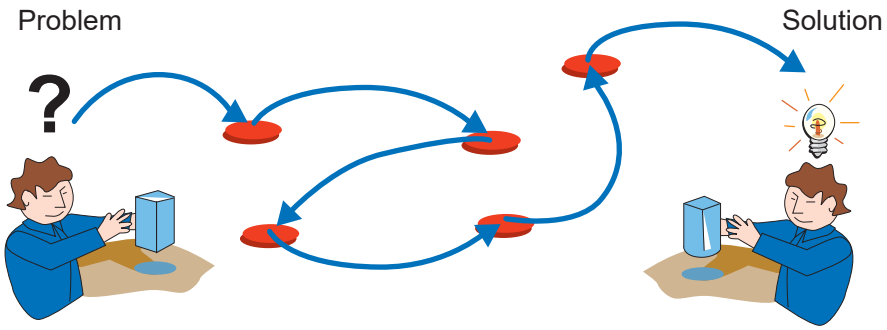
- Re-use
- Re-fine
- Re-duce
- Re-inforce
- Re-form
- Re-place.

Do you rely on luck, or are you destined to win based on excellence?

7.9 Concept development

Few areas are surrounded by as many myths and





Rephrasing the problem by moving it into the functional domain.

falsehoods as concept development and innovative problem-solving. Here is an excellent explanation made by the Triz master Yuri Salamantov of what a creative solution is:

- *"The creative solution is a simple solution to a problem that, from the standpoint of common sense, would seem utterly complex."*

The above quotation beautifully captures the challenge facing many development engineers. They can come up with solutions, but the crux is that the solutions are too complex, expensive, or difficult to implement. Most technical problems can be solved if unlimited resources are available or if there are no constraints. However, the creative solution is so simple that it may almost be embarrassing: "*Why didn't I think of that?*"

A common myth is that the creative solution appears like magic out of thin air and that structure and systematics are the direct antithesis of creativity. Instead of spending time understanding and rephrasing the problem, the first feasible solution that comes up is accepted. While it's true that some problems can be solved through wild brainstorming. Occasionally, you can have luck, but it will not provide great results in the long run.

The previously discussed tactics provide a general description of possible ways to modify the functional model of your benchmark to enhance customer value. They represent different avenues to explore to address your value-increasing challenges. Some of these challenges are straightforward and can be resolved using standard engineering tools. A few must be included in your research portfolio and are not part of product development. However, the challenges that lie in between are the ones you need to tackle in your development project.

At first glance, these seemingly complex challenges require more powerful creative tools than just wild brainstorming to resolve.

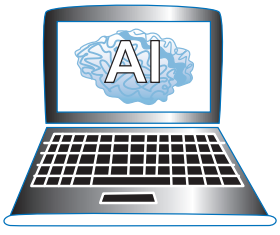
Fortunately, several powerful creative tools are available, and a few will be introduced here:

- AI-based tools
- Triz tools
- Focused brainstorming

The value-increasing challenges formulated in the Functional Domain of Information are ideal for leveraging artificial intelligence (AI) technologies. You have skillfully crafted questions that AI tools can comprehend and process effectively.

As of 2025, I have tested several available AI tools, and the results are promising. The rapid advancement in this field is remarkable, and I am convinced that we will witness exponential growth and development in AI capabilities.

An AI tool capable of tapping into the world's patent database, understanding the technical language, and transferring patented and protected solutions from one category to unprotected categories aligns with the fundamental idea behind patents. Contrary to popular belief, patents don't solely exist to protect innovators' intellectual property. Instead, patents serve as a mechanism for innovators to be compensated for sharing their ideas with humanity. In essence, patents provide a temporary monopoly on an invention, allowing the innovator to recoup their investment and profit from their crea-



tion while ultimately contributing to the greater good.

All AI tools I have tested also quickly became my subject matter of expert. Someone I could chat with and explain different limitations of the technology, technical aspects, and physical principles. My recommendation is to give it a shot. You might be positively surprised.

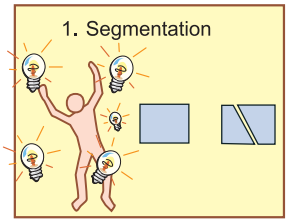
Triz is the Russian abbreviation for Theory of Inventive Problem Solving. This technique has its roots in Russia and the fundamental work done by Genrikh Altshuller. The word method is wrong as it is more of a toolbox with many different methods and tools. TRIZ is the science of how technical systems develop towards becoming more ideal, that is, having higher customer value. Genrikh Altshuller's innovative approach involved analyzing patents from a unique perspective. He sought to understand the problems patents aimed to solve and the underlying principles used to resolve them. Despite the vast number of patents, Altshuller discovered that the fundamental solutions could be condensed into a manageable number. His work resulted in three essential components of the TRIZ toolbox:

- The 76 Standard solutions. The standard solutions address common problems defined by their Su-field models. A Su-field model is a different type of functional model than the one I describe and use. This approach provides a structured way to resolve recurring technical challenges. You can find plenty of material on the Internet if you want to learn and master the standard solutions.

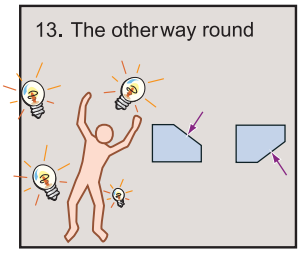
- 40 Principles of solutions to typical technical problems: These principles represent a distilled set of solutions to common technical challenges. These are excellent for focused brainstorming, which will be described later. In my Triz road map, which can be downloaded at the site, these 40 principles are listed.

- Ariz is a problem-solving algorithm. Based on the Ariz, I have created a Mini-Ariz, a simplified version of the full Ariz. It is less powerful and sharp when solving technical challenges. Its advantage is that it is more intuitive and easier to learn and master. My Triz road map explains the mini-Ariz, which can also be downloaded at the site.

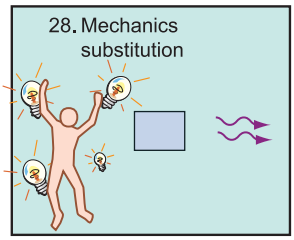
I am not against brainstorming but firmly against wild brainstorming, especially with the goal of thinking "*outside of the box.*" Going wild and thinking outside of the box produces mostly brain farts. It is an illusion of creativity that sometimes makes people feel good, but it is a waste of time. The process gets cluttered with too many low-value and silly ideas, ideas you have to evaluate and eventually kill.



Instead, you should always brainstorm inside a new, clearly defined box. You need to set strict boundaries for the type of ideas allowed.



The 40 Principles of Solutions to typical technical problems is a perfect example of how you can define a new box. Each principle represents a clearly defined new box. Or use analogies like biomimicry may also work.



The more boundaries you set around what type of solution you are looking for, the harder it will be to develop an idea. By forcing people into a new box people have never considered, you force them to "**think outside the existing box but inside a promising box.**" In a box with a higher probability of finding a valuable solution.

The more you apply the tools described above, the quicker and better you'll become at using them. The following additional guidelines should also be followed as much as possible:

1. Use simple language and remove all technical expressions as they block the process.

2. Let the ideal system be a guiding light. A product that only performs Main and Additional functions at minimal resource consumption.

3. Build functional models to define all the problems and create a united picture of the context. Sometimes more than one. One on the customer level, another on the technical level, or why not on the supersystem or physical level?

4. Amplify the problem to extremes. If there is a problem at 100 rpm, increase the motor speed to 10,000 rpm or more and look for solutions.

5. Clarify the conflict zone. Where is the problem? Go from macro to micro and sometimes even to atomic or molecular level.

6. Clarify when the problem arises and what happens before, during, and after. What happens before, and what can be done before? What happens when the problem is active, and what can be done during this time? What happens afterward, and what can be done afterward?

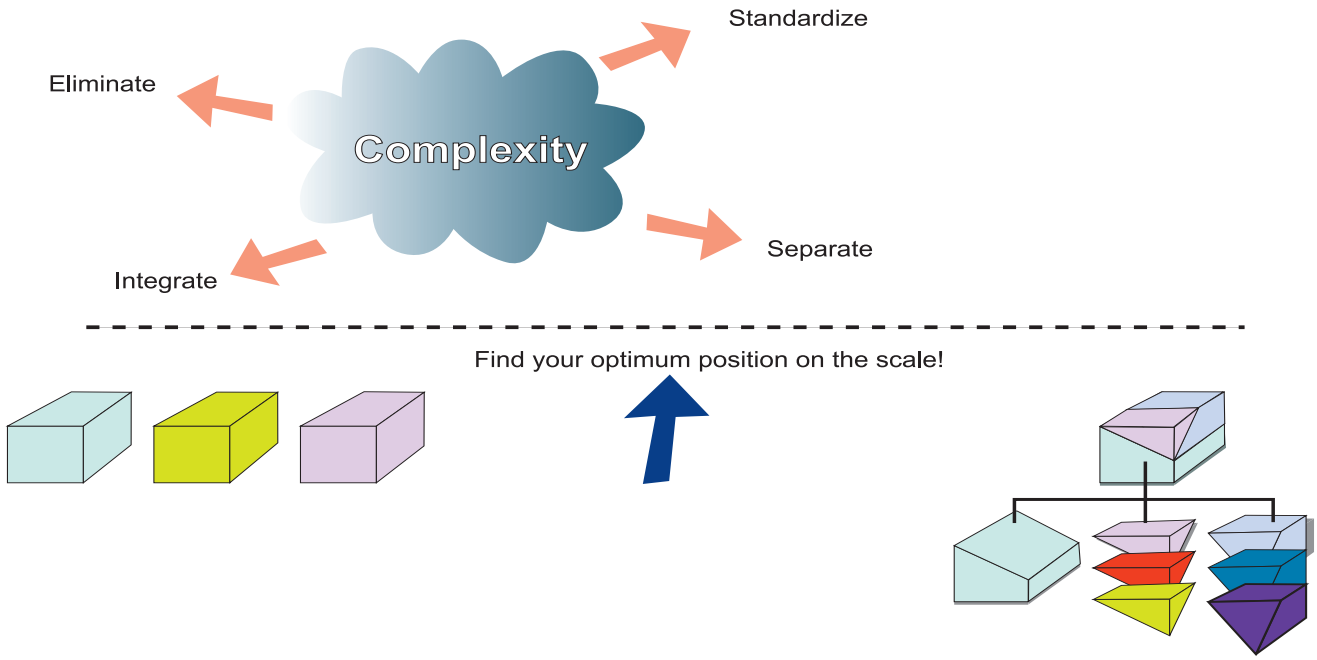
7.10 Establish the System architecture

The unspoken rule seems to be that when products become more complex and include people processes or software, they start to be called systems. Therefore, I will use the term system architecture, which is equivalent to product architecture.

System architecture has a profound impact on a project's success. It can guide the project and business case around potential pitfalls or lead them straight into trouble. A poorly designed system architecture can be the source of many problems, costs, and headaches, both during the project and later when customers start using the product. However, identifying system architecture as the root cause of these issues can be challenging. Typical problems with the root cause of poor system architecture include:

- a complex development project
- many late changes
- supply chain challenges
- high production costs
- manufacturing problems or difficulties
- customer complaints.

High standards must be set on the system architecture. It is not unreasonable to demand the creation and evaluation of several alternatives. The system architecture is most easily made using a comprehensive functional model of your final concept.



Complexity is the major villain that separates an excellent system architecture from a poor one, a villain that appears in many guises. Therefore, the most important goal when designing the system architecture is to minimize complexity from a holistic perspective. However, the four best strategies for mastering complexity pull in different directions. They are largely each other's opposites. The difficulty, therefore, lies in creating the perfect balance between these conflicting strategies. The four strategies are:

- **Eliminate**: where can elimination create advantages?
- **Integrate**: where can integration create advantages?
- **Separate**: where can separation create advantages?
- **Standardise**: where can standardization create advantages?

The critical issue is maintaining a holistic perspective and viewing costs at a consolidated level. Using a more expensive component that, for example, doesn't emit heat increases the costs of one module. Still, it may reduce the cost of another module or completely eliminate the need for it. In large organizations, such a change may be challenging to implement, as each department or project team only are responsible for the costs of their own module.

An example of integration is designing subsystems and components so that they can perform more functions and thereby eliminate other subsystems or components. This increases the complexity of one module but reduces the overall complexity of the whole system. Look for functional elements that don't need to be retained independently of others. These types of elements are always candidates for elimination. One method suitable for mechanical systems that investigates the possibility of integrating elements is Design for Manufacturing and Assembly (DFMA).

Separate, for example, critical or complicated subsystems or components so that these parts of the product can be developed independently from the rest. Different internal or external teams can develop these subsystems or components simultaneously, which is especially important with subsystems undergoing speedy technical development. In other cases, separation can increase customer value by simplifying the replacement of components that wear out.

A third case can be that it simplifies the manufacturing or supply chain. Interfaces must be defined to manage any eventual dependency between the separated components. These interfaces must be standardized and kept holy to ensure the different elements can be developed parallel and correctly assembled.

Standardize subsystems, modules, and interfaces or adapt them to fit already established industry standards to increase flexibility within the product. For example, many physical products can reduce the number of unique brackets, contacts, cables required, etc.

Twist and turn these four strategies and take them to the extreme. How can the system architecture be changed to:

- Maximize customer value? What should then be eliminated, integrated, separated, and standardized?
- Minimize complexity in the product? What should then be eliminated, integrated, separated, and standardized?
- Simplify production and distribution? What should then be eliminated, integrated, separated, and standardized?
- Reduce demands on development work? What should then be eliminated, integrated, separated, and standardized?

In this way, a range of different alternatives are produced. Forcing each strategy to its extreme expands the investigated solution space and releases a sphere of possible system architectures within which the optimal compromise exists. The viable option is usually between integral and modular architecture extremes. Both extremes have advantages and disadvantages.

I prefer to utilize a comprehensive functional model of the final concept for this exercise. It is essential to identify and circle those functional providers that should be included within the

same subsystem or module. Clearly articulating the advantages of integrating them into the same subsystem or model is essential. Additionally, analyzing the functional implications is crucial when defining the final system architecture. The greater the number of subsystems and modules required to collaborate for a Main or Additional function, the more complex the design and manufacturing processes become. Low complexity is what separates an excellent system architecture from a poor one.

7.11 Detailed design of the final product

Most engineering hours and costs are usually consumed in this phase. The work is product-, industry- and company-specific. The product and process intertwining make the area extensive and complex. The detailed design of the product must go hand in glove with the design of the manufacturing and distribution processes. Cross-functional work methods must be the foundation on which the detailed design of the product and all its processes is built. This area falls within the core competence and expertise of your own company. Therefore, I would be presumptuous to believe I could give concrete advice on implementing this work. Your company must provide this expertise. If you are required to make a Design FMEA, the functional model of your final concept is your best guide. You can learn more about that in the Functional domain of information.

We are also moving away from value-critical information to specific or supplemental information. Specific information makes up the majority of information to assist you in advancing along your chosen path.

7.12 Summary

The Solution Domain of Information represents a unique world within your company, a space where innovation is not only possible but entirely within your grasp. Here, you have the power to decide on the development of products and services that resonate with both your vision and your customers' needs. However, organizations frequently get bogged down in an ineffective mindset, prioritizing countless minor tweaks and ideas rather than embracing meaningful innovation. When organizations become preoccupied with trivial tweaks, they inadvertently close themselves off from exploring groundbreaking ideas that could redefine their industry.

To combat this detrimental mindset, the chapter outlines several actionable strategies to foster a meaningful innovation culture. Central to these strategies is implementing a structured approach that incorporates powerful creative tools, which serve as catalysts for transformative thinking.

Value-critical information in the Solution Domain of Information includes:

- Careful selection and use of several of the six tactics to modify the present benchmark to enhance its customer value. It will probably start with the re-use and include the re-duce tactic.

- The modification of your present product or the benchmark is solving a substantial number of the value-increasing challenges formulated in the Functional Domain of Information. The final concept has the potential to have unrivaled customer value.

- Several powerful creative tools have been used in concept development, including AI, Triz, and focused brainstorming.

- A system architecture that reduces the complexity of development, supply chain, manufacturing, and customer service. Carefully balancing the incompatible strategies of Eliminate, Integrate, Separate, and Standardize.

- A structured FMEA based on your final functional model.

- Leveraging the learning curve to drive down costs.

