BEST PRACTICE 2.3

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.

The second problem is that VUCA leaves out two additional forms of uncertainty: Incompleteness and Black Swans, also shown in Fig. 1. A better acronym could be VUCA+I+BS, but this is too cumbersome. Therefore, VUCA+ is used in this book.

Why this concern for uncertainty? Virtually every piece of information is uncertain. This is true in all aspects of an evolving product: the requirements, the engineering specifications, the dimensions, the electronic components, and the code. No amount of engineering can eliminate all the uncertainty, but successful organizations try to understand and manage it. They know when to spend resources to reduce uncertainty. Best practices aim to make uncertainty visible and manageable.

In the following sections, each of the elements of VUCA+ is defined.

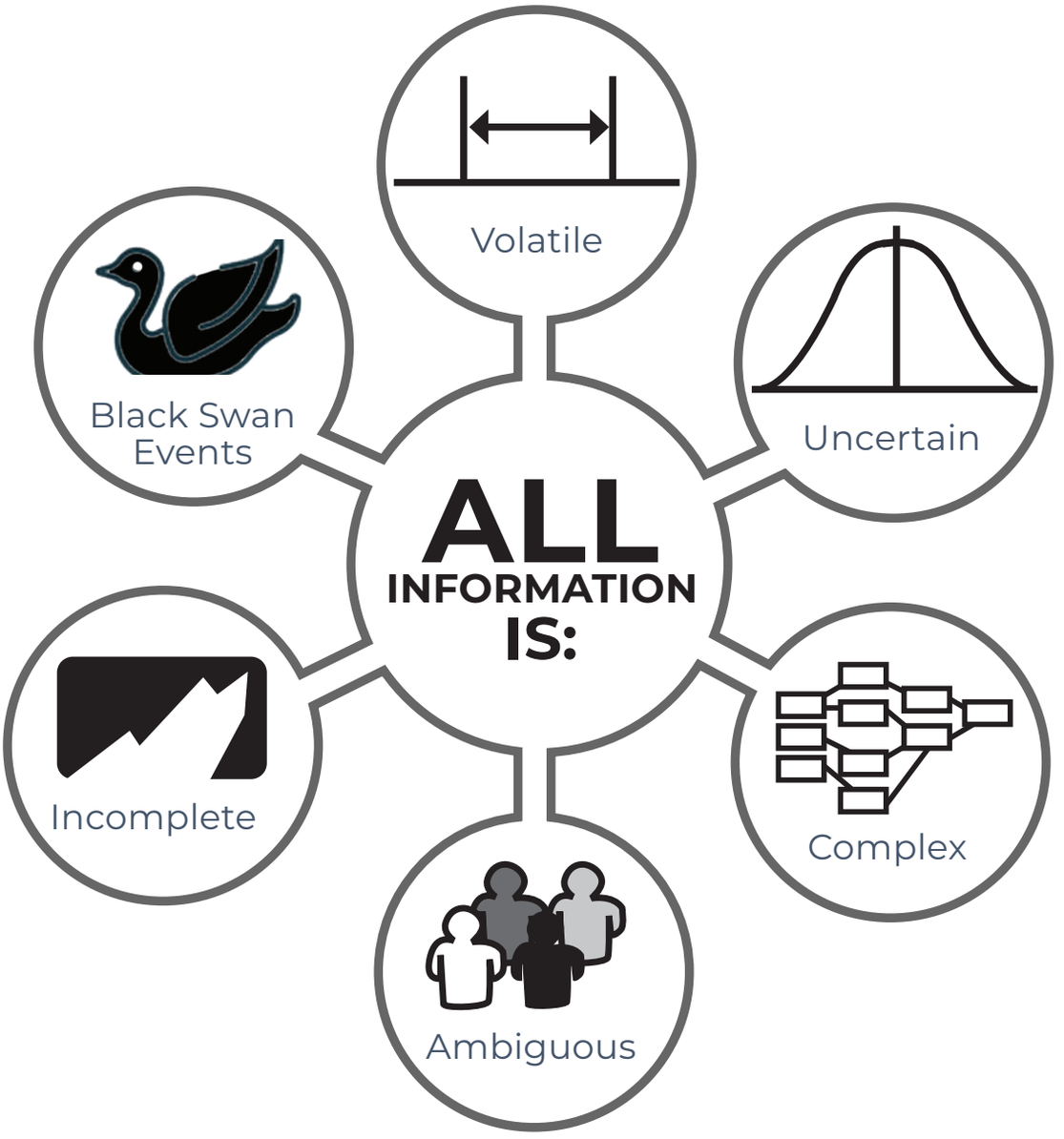
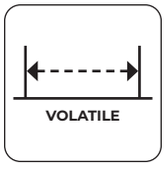


Figure 1. The elements of uncertainty or VUCA+.



"V" is for Volatile

Volatile information is changing and evolving.

The "Volatility" icon in Fig. 1 represents the range that an important value might have during the design process. Specifications, dimensions, voltages, user interface elements, computer code, and materials all evolve and change as an inevitable part of the design process.

There are degrees of volatility along a spectrum from high (rapid and unpredictable) to low (infrequent and predictable). The issue is whether important variables are changing to the extent that they may affect the design process. If they are, then it is crucial to understand and control them.

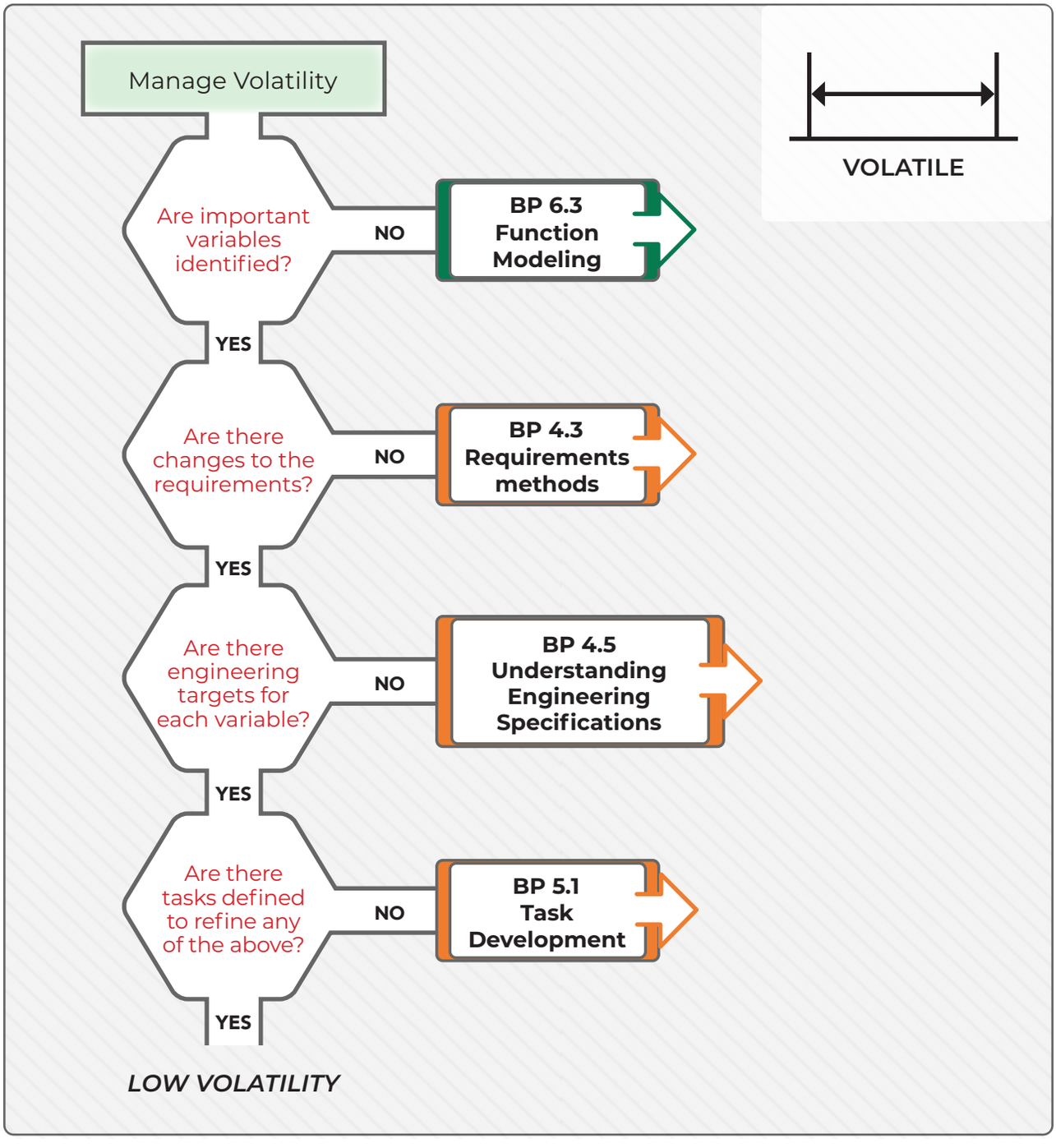


Figure 2. The logic for managing volatility.

The logic for managing volatility is shown in Fig. 2. The first step is to identify critical variables. Successful engineers will identify variables to be concerned about at various times in the design process. Generally, variables of concern affect function, and functional modeling is one method that helps engineers determine critical variables.

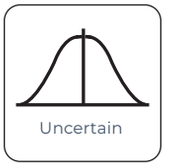
The requirements must be known for each important variable. Requirements and the engineering specifications developed from them help manage volatility by establishing targets. The specifications themselves may be volatile because the target may be moving, and a moving target is generally bad because it may make the project impossible to complete.

Periodically, new information will arise that compels organizations to revisit specifications. For example, if a competitor introduces a product with better performance than the specification requires, it may be prudent to change the specification on some features.

If specifications are firm, engineers can reduce volatility by working on tasks to refine variables to meet them. If a specification states that the weight must be less than 50kg, engineers should perform weight assessment and reduction tasks to determine whether this target value is met.

Surveys of product development organizations have revealed five primary reasons for volatility:

- Fixing defects – previous values were not providing the desired result, so the specifications need to be changed.
- Missing information – something important was overlooked.
- Functionality enhancement – a new or improved function was added, which is so common that it is called “feature creep”: the previous functional requirements are expanded, usually extending the development time and cost.
- Change in product strategy – a change by marketing impacted the product’s functionality; this may be a Black Swan that surprises the design team.
- Design improvement – new knowledge allowed for an improved product.



“U” is for Uncertain

Uncertainty is a lack of precision or a variation in value. In statistics, the standard deviation about the mean is a way to represent variation and uncertainty: the more uncertainty, the higher the standard deviation. For example, materials vary in properties and power supplies provide an unsteady current. Knowing and managing the variation in them is crucial during design.

Every variable used to describe a product has a variation, whether for a physical property, lines of code, or the number of steps needed to use the product. Some variables, and their effect on function or manufacturability will be important at any point. As shown in Fig. 3, there are four questions to ask of each of these variables. First, is the variation known? It is considered insignificant if the variation is known and lies within a range that does not adversely affect functionality, manufacturability, or assembly. If the variation is unknown, a task, generally a simulation or test, must be performed to learn more about it.

Second, if the variation is not known, then the question is how to get that information. Once the variability is understood, then it is time to judge whether the variation will affect the function.

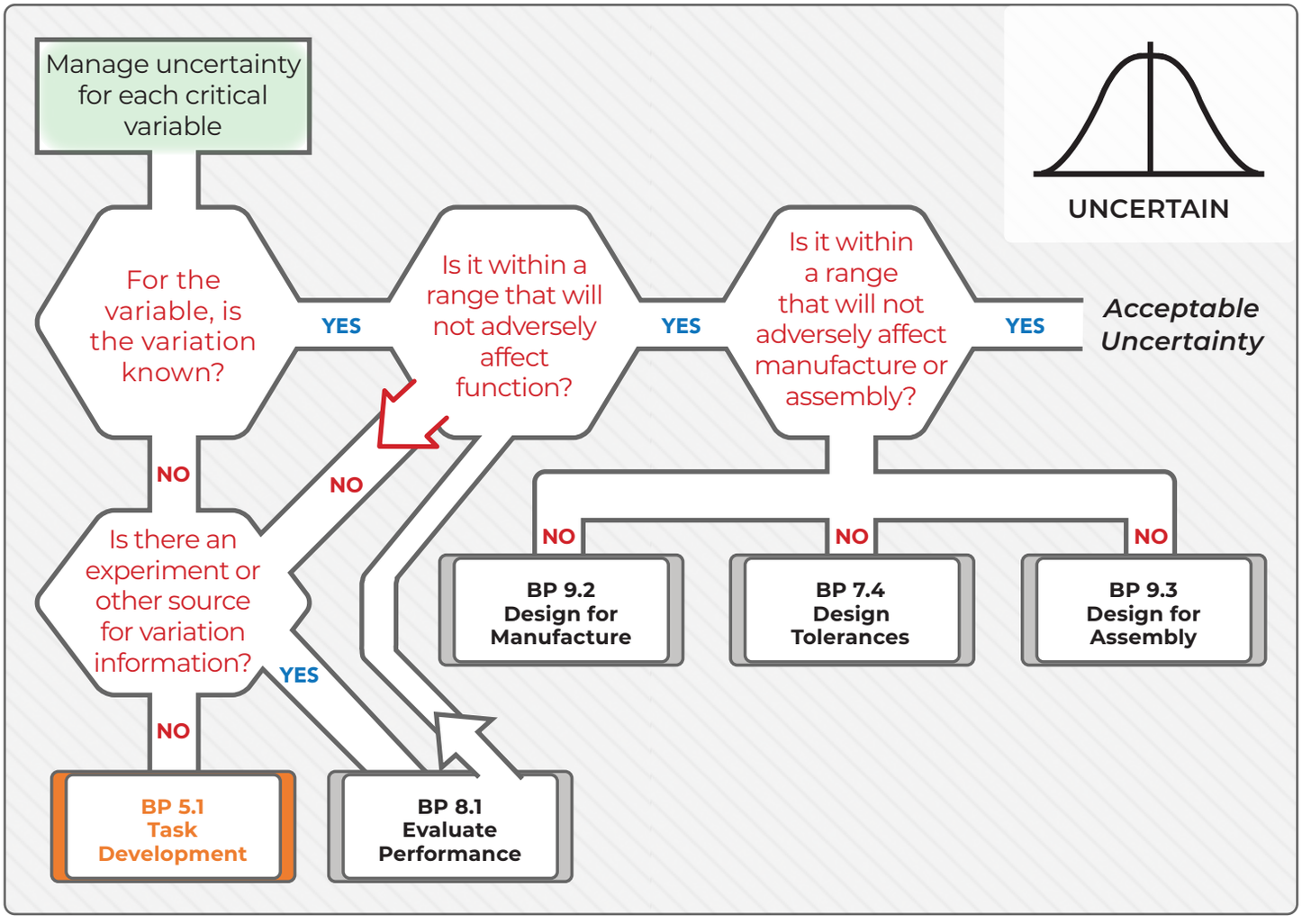
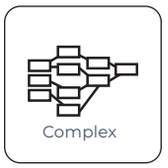


Figure 3. The logic for managing uncertainty.

Finally, is the variation in a range to make manufacture and assembly economical and efficient? If not, then the manufacturing method, tolerances, or assembly procedure may need refinement.

In Best Practice 7.4, knowing the standard deviation allows engineers to compute the probability of the variation being an issue. Successful organizations use probabilities to manage uncertainty based on historical or measured data. This is different from working with possibilities, which are guesses used when good data is unavailable.

Most design problems are redesign problems since they are based on prior, similar solutions. Conversely, most design problems are original as they contain something new that makes prior solutions inadequate.



"C" is for Complex

Most modern systems are complex. A decision on one variable may provide new or changed information that impacts other systems. While in an ideal world all the variables would be independent, this is rarely the case. Good design practice aims to help the designer manage this complexity (see Fig. 4).

Successful organizations have found that carefully designing the interfaces between systems is key to managing complexity. They found that these boundaries or interfaces must be fixed and stable. This means that the transfer of information, materials, energy, and control (the functionality) between systems 1 and 2 will need to be well-defined and only changed with careful forethought. The flow of questions and best practices in Fig. 4 enables this design philosophy.

Successful organizations take care in defining systems, the functions provided by them, and the interfaces between them.

> **"Fail early; fail often; repeat."**
>
> —CHRIS BOUWMEESTER

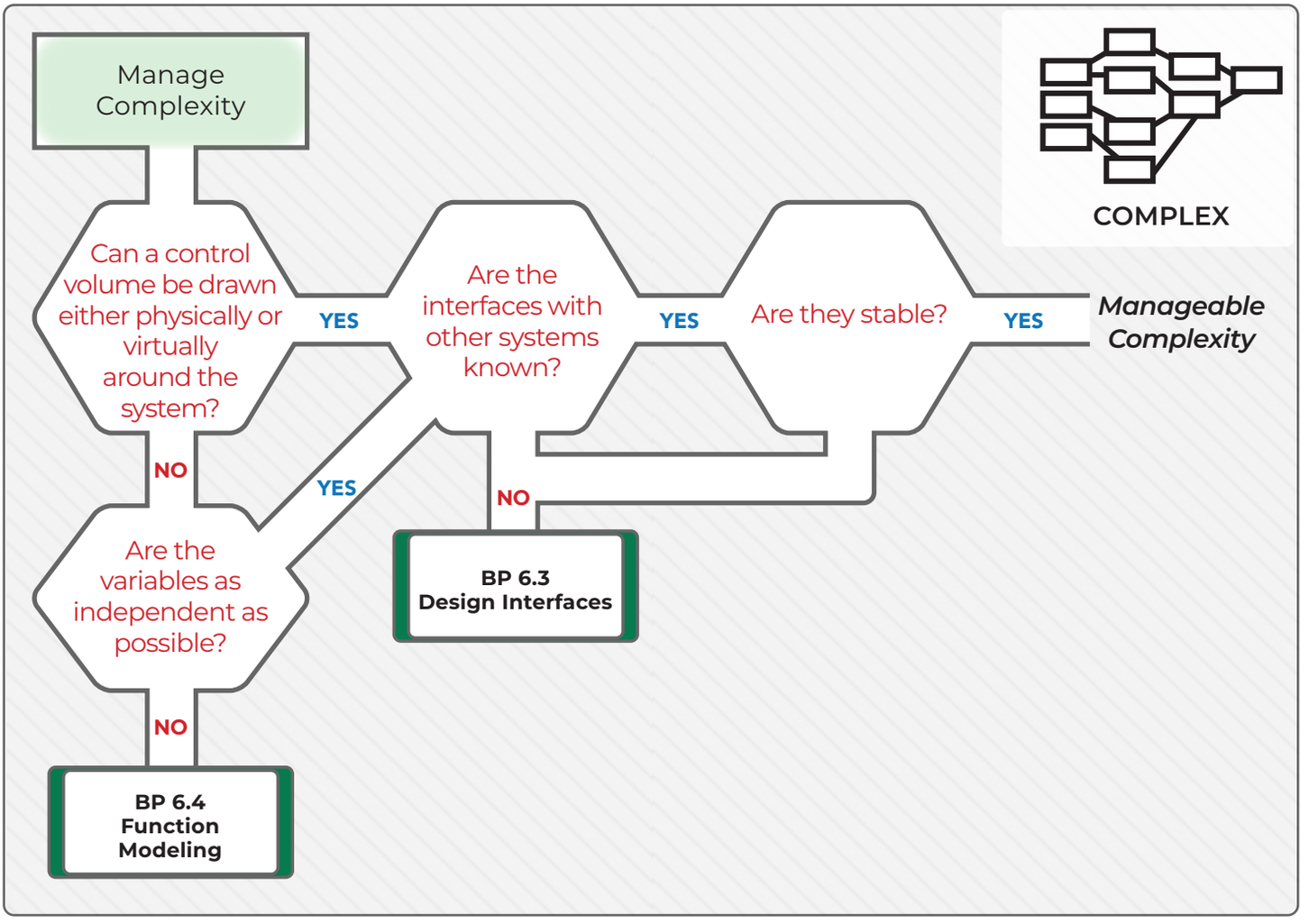


Figure 4. The logic for managing complexity.



"A" is for Ambiguous

Ambiguous information can be understood or interpreted in more than one way. When an object is evolving, regardless of whether it is a complex system, a simple feature on a part, or a line of code, it is ambiguous until it is finalized and documented. This is natural as everybody makes sense of abstract concepts based on their unique experience and knowledge.

If managed well, ambiguity can lead to a better design as the team works to evolve the product. This is where a strong, diverse team environment can be an asset. Such an environment allows engineers to see complex problems from different viewpoints and ensures that ambiguity is not a factor in developing a shared vision.

The questions and best practices in Fig. 5 improve team communication and the development of a shared understanding about any system, subsystem, snippet of code, component, or feature.

It is probably most important to remove the ambiguity with the customer. If the designers and customers do not see the product the same, there is a significant disconnect. Ambiguity may not be resolved until an object is reduced to a working version.

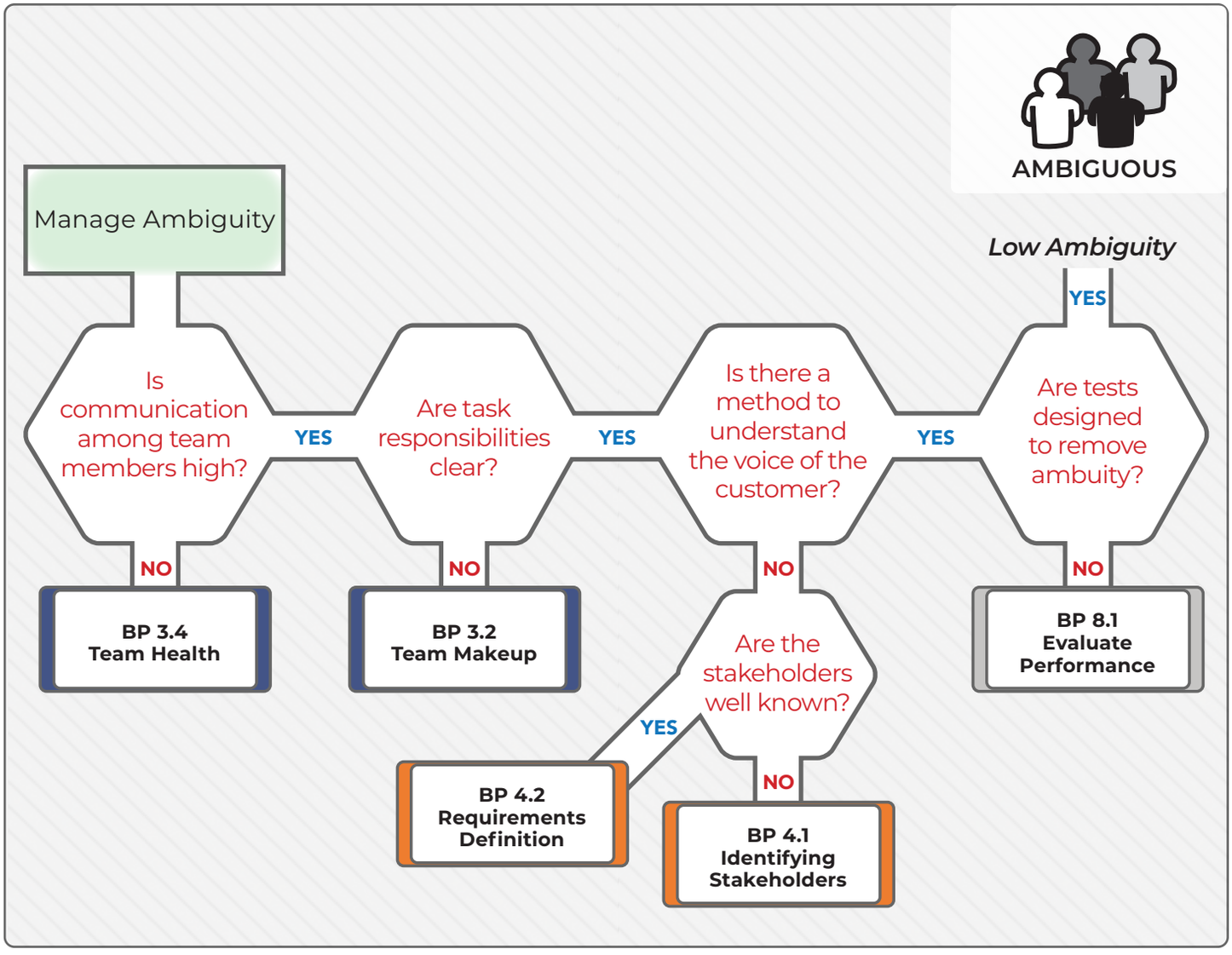


Figure 5. The logic for managing ambiguity.



"I" is for Incomplete Information

During design, information is always missing. Incompleteness is reduced whenever a mechanical designer adds a dimension in a solid modeling or CAD system, an electrical engineer specifies the need for a resistor and its value, or a programmer adds a new variable and writes code.

There are distinct kinds of completeness:

- Variable completeness: Are all the important variables that significantly affect functionality and performance known?
- Specification completeness: Is there an engineering specification for each variable that includes a target and threshold? Is the team aware of when this specification is needed?
- Design completeness: For each variable, does there exist a value and tolerances? If they are present, are they final (documented and most likely unchanging)?

Smart organizations do not strive for completeness; rather, they put significant effort into identifying the critical variables that affect functionality and develop specifications, dimensions, and code as needed. This is the logic shown in Fig. 6, complete with links to best practices used to provide the needed information.

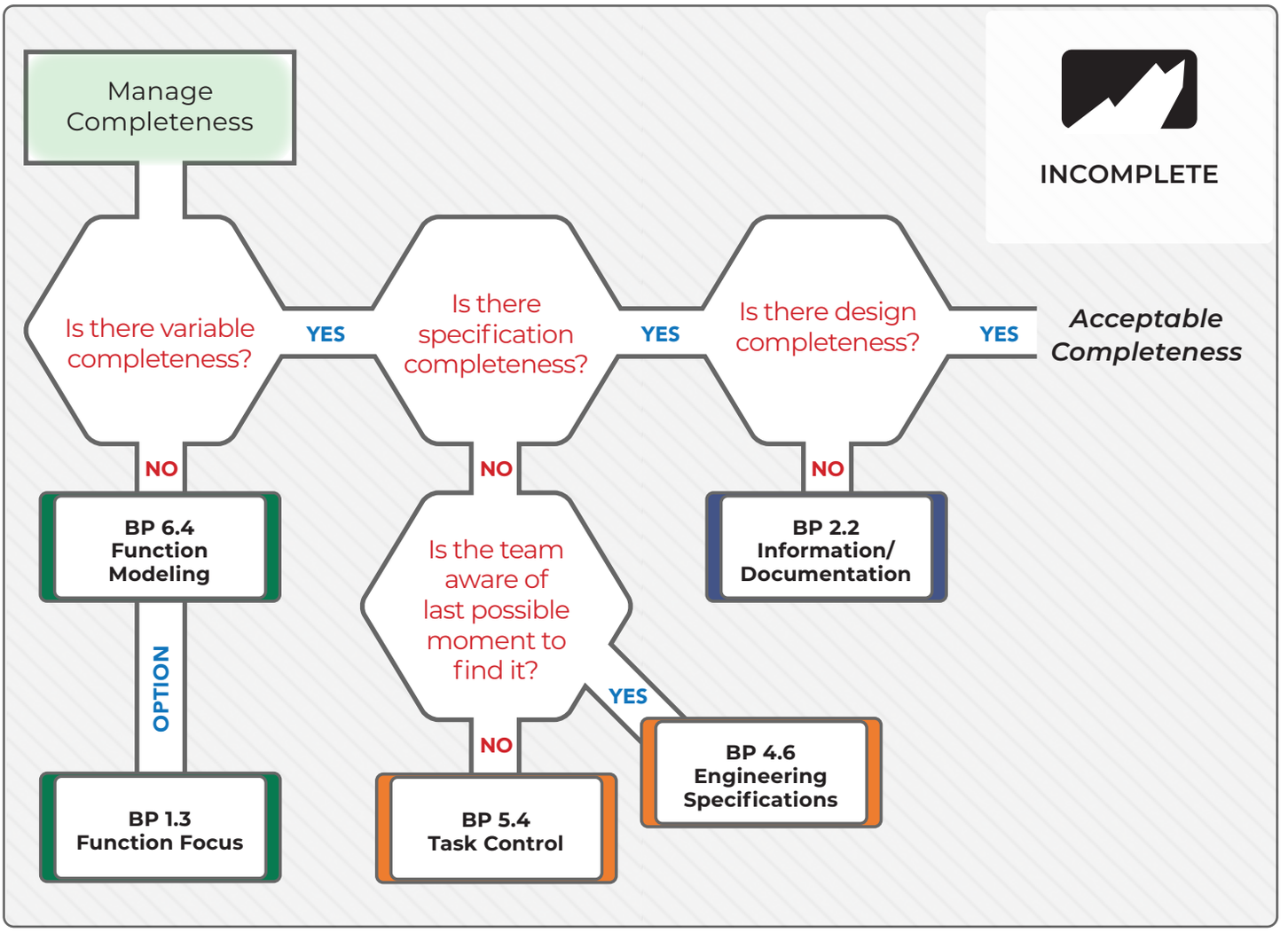


Figure 6. The logic of completeness.



"BS" is for Black Swans

Black Swans are unforeseen events that have a major effect on the product. The name comes from the 17th century when Dutch explorers first saw black swans in Australia. While white swans were common in Europe, no European believed black swans could exist, and so the term came to mean a perceived impossibility that might later be disproven.

On a national level, the 9/11 attacks on the World Trade Center are generally considered as Black Swan events; few thought them possible until they occurred. Black Swans events have three attributes:

1. The event is unpredictable (to the observer).
2. The event has widespread ramifications.
3. After the event has occurred, people will assert that it was indeed explainable and predictable.

The first attribute is very fragile and explains the caveat "to the observer" because a wise or learned observer may have been able to foresee the event. In the case of 9/11, for example, airplanes had flown into buildings

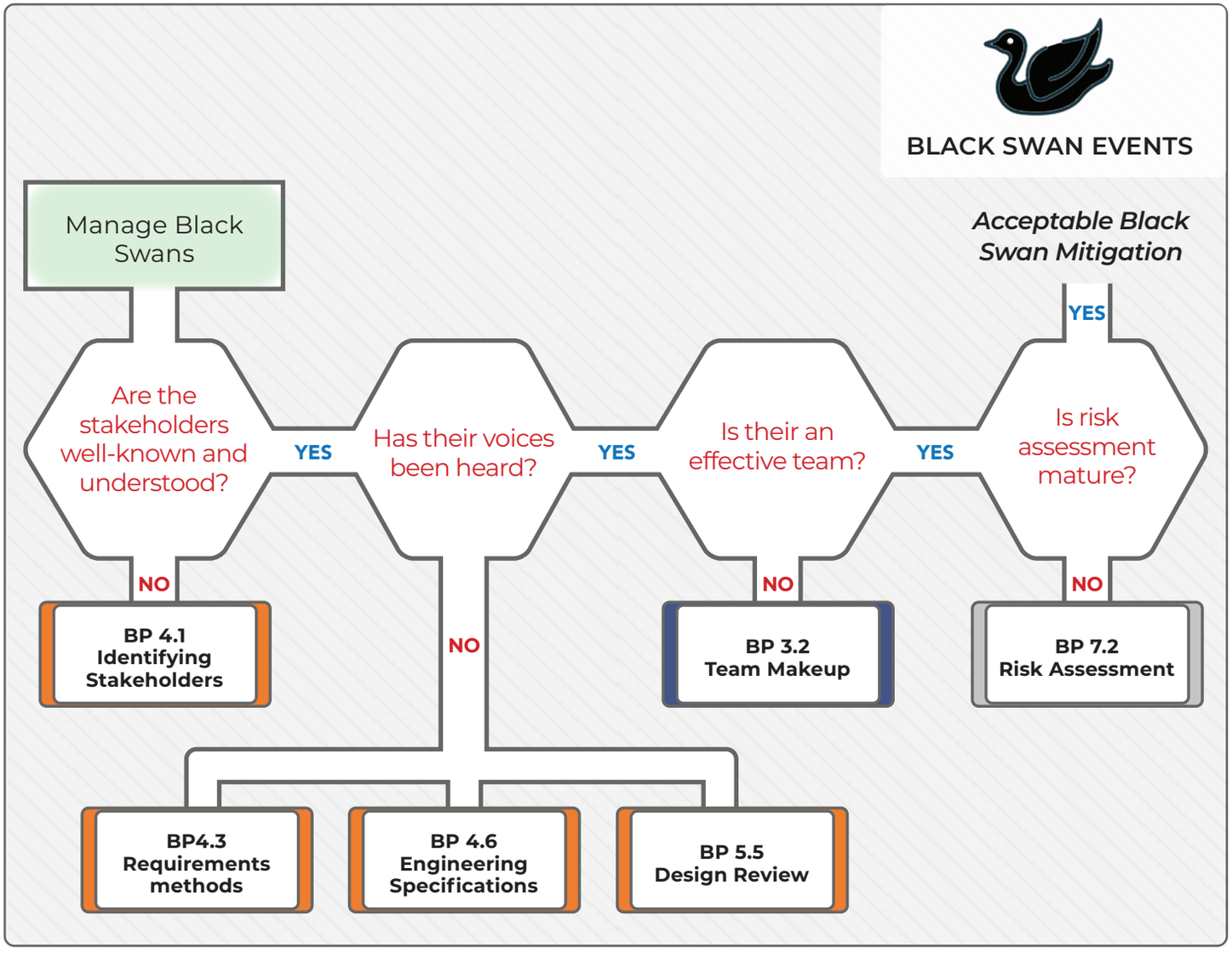


Figure 7. Black Swan reduction logic.

before, even in New York City. In fact, in 1945, a B25 bomber hit the Empire State Building in fog only blocks from where the World Trade Center was hit on 9/11. This was an accident, but there has been other, on-purpose, ramming of buildings. Further, the hijackers of the 9/11 airplanes received flight training with little concern for landing, which is usually the hardest part of learning to fly an airplane. This evidence could have affected the event's predictability, and a learned observer with the right information could have foreseen the event. This may just be a case of the third attribute in the list above.

For product design, some Black Swans are:

- A competitor unexpectedly announces a product that changes the market for your evolving product.
- A significant change in organizational management.
- The product is canceled (or drastically changed) by management.
- A critical component (or material or sub-assembly) supplier cannot deliver as planned.

One goal during the design process is to foresee as many Black Swan events as possible and compensate for them in the design plan. Successful organizations work to be risk-aware by making every effort to predict what can go wrong, having solid and multidisciplinary teams, and using many of the best practices in this book.

Risk is defined as 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?

Best Practice 2.3, Manage Information Uncertainty and Risk VUCA+ Worksheet			
			For instructions, hover here.
Project:	**Requirements Authoring Tool**		
System/Topic	**Requirements Guided Authoring Excel + VBA**		
Engineer:	**Shanae E.**	**Date:**	**12/01/23**
Version of Worksheet:		**2**	
Volatility			
BP 2.3, Figure 2	**For a system, sub-system, assembly, component, code, or feature of any of them:**	**Answer**	**If "No" see Best Practice**
V1	Are important variables identified?	No	**6.3, Design from Known Stable Interfaces**
V2	Are there changes to the requirements?	Yes	
V3	Are there engineering targets for each variable?	No	**4.5, Dissect Products to Gain Product Understanding and Find Market Opportunities**
V4	Are there tasks defined to refine each of them?	Yes	
Uncertainty			
BP 2.3, Figure 3	**For any critical variable in the system:**	**Answer**	**If "No" see Best Practice**
U1	Is the variation known?	Yes	
U2	Is there an experiment or other source for variation information?	Yes	
U3	Is it within range that will not adversely affect function?	Yes	
U4	Is it within a range that will not adversely affect manufacturing and assembly?	Yes	
Complexity			
BP 2.3, Figure 4	**For a system, sub-system, assembly, component, code, or feature of any of them:**	**Answer**	**If "No" then see these Best Practices**
C1	Can a control volume be drawn either physically or virtually around the system?	No	**6.2, Develop and Explore Multiple Concepts in Parallel**
C2	Are the variables as independent as possible?	Yes	
C3	Are the interfaces with other systems known?	Yes	
C4	Are the interfaces stable?	Yes	

Figure 8. VUCA+ Worksheet Requirements Authoring Tool Example.

2. How likely is it to happen?
3. What are the consequences if it happens?

Risk is a direct function of uncertainty. Some degree of uncertainty is part of nature and cannot be controlled (e.g., the weather, earthquakes, material, and manufacturing variations, etc.). However, much of the uncertainty exists because of a lack of knowledge during conceptual design. If everything is known precisely, a product can be designed with little (or at least known) risk. Unfortunately, incomplete knowledge, low-fidelity simulation results, manufacturing and material variations, and unknowable acts of god all contribute to risk.

Successful organizations use risk management to reduce the impact of the unanticipated threats using the four questions in Fig. 7. These questions ensure that surprises are minimized when an effective team is tuned in to hear the voices of the stakeholders and are actively doing risk assessments.

The VUCA+ Worksheet

The worksheet sample in Fig. 8 summarizes this best practice as a checklist. The sample here only shows Volatility. The indentations of the questions coincide with the structure of Fig. 2. The complete worksheet has all the logic in Figures 2-7 included in it.

Related Best Practices

Term	Definition	Best Practice
Function Model	Black box models that explain the functions associated with sub-systems and components. An important early step in concept development.	6.4
Requirements or (Design Requirements)	What the object should do, a characteristic of it, 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
Specification or engineering specification	A formalized requirement includes a subject, units, target, and thresholds.	4.6

Resources

Nurmuliani, N; Zowghi, D. and Powell, S. "Analysis of Requirements Volatility during Software Development Life Cycle." *2004 Australian Software Engineering Conference 13-16 April 2004, Melbourne, VIC, Australia.* Edited by Paul Strooper, IEEE, 27 September 2004. DOI: 10.1109/ASWEC.2004.1290455

Andreux, Dominique. "VUCA: The New Reality" *QRM Institute*, https://qrminstitute.org/vuca-the-new-reality. Accessed 2023.