

# WHITE PAPER

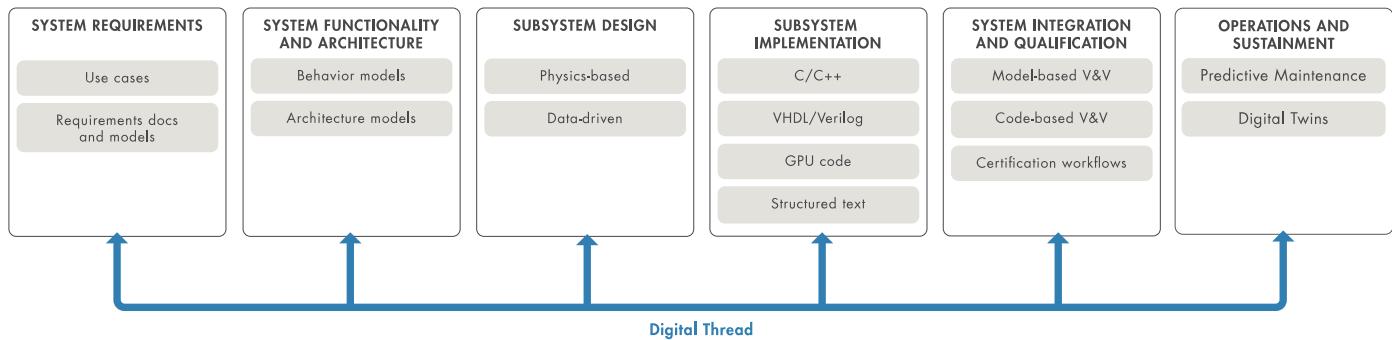
## How Engineering Teams Adopt Model-Based Design

Regardless of the size of the company, Model-Based Design always starts with a small engineering team and is adopted to reduce development time and costs. This paper outlines the successful adoption practices and considerations used by small teams.



## What Is Model-Based Design?

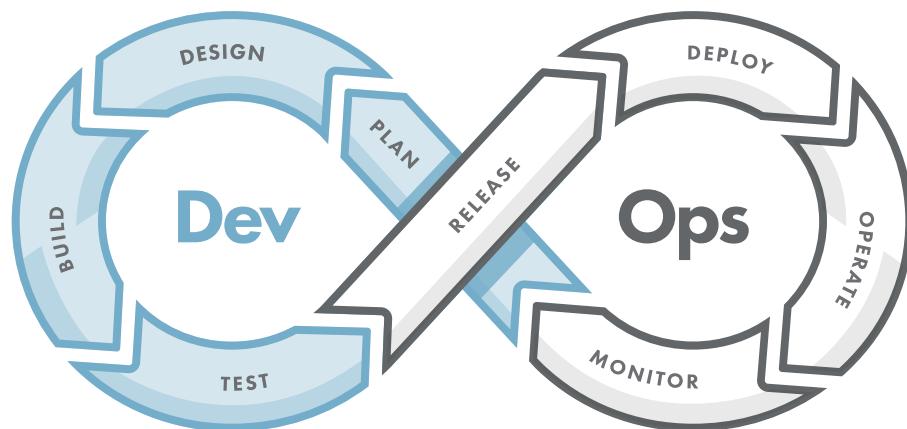
Model-Based Design provides a mathematical and visual approach to develop systems. It is used across a wide variety of industries and applications including motion control, signal processing, industrial equipment, and aerospace and automotive applications. Model-Based Design centers on the systematic use of models throughout the development process for requirements specification and validation, design, analysis, simulation, automatic code generation, and verification and validation.



*Model-Based Design: Systematic use of models throughout the development process.*

Model-Based Design complements and enables Agile development practices. Like Agile, Model-Based Design enables developers to discover defects and design problems in the early stages of a project, accommodate changes in requirements, and deliver a system that meets customer needs. For example, users can perform fast iterations by connecting with continuous integration (CI) systems to automatically test and verify models and code throughout the development lifecycle.

Users can further extend the use of previously developed models to the systems in operation, creating digital twins to optimize system operation, monitor system status, and provide a feedback mechanism to the development team to make continuous improvements. This DevOps approach enables applications such as predictive maintenance and real-time fault detection.



*DevOps: A set of practices to automate and integrate processes between Development and Operations.*

Engineers create a model to specify the behavior of a system. The model, which consists of block diagrams, other graphical elements, and textual programs, is an executable specification that lets engineers run simulations to test alternate design concepts and verify designs throughout the development process.

The benefits of having a system model serve as the executable specification include:

- The design can be tested, refined, and retested throughout the development process.
- Modeling and simulation enable engineers to quickly try many ideas without the need for expensive prototypes.
- Testing and validation are done early and continuously rather than at the end of the process so that errors are found and corrected before hardware testing.
- Code can be generated from the system models, which reduces effort and eliminates hand-coding errors. This code can be used for testing real-time simulations and deploying to hardware.
- Models can be adapted and reused on subsequent projects.

## Getting Started with Model-Based Design

Teams cannot afford to be slowed down by changes in the way they develop systems. In practical terms, this means that any new tools and process changes need to be introduced carefully. Trying new approaches and design tools carries an element of risk. Successful teams have mitigated this risk by introducing Model-Based Design incrementally. They usually start with a single project, identifying early wins that can be achieved using Model-Based Design versus using the current practice.

Successful introduction of Model-Based Design involves taking incremental steps that can help a project along without slowing it down:

1. Experiment with a small piece of the project.
2. Build on initial modeling success.
3. Use models to solve specific design problems.
4. Stick with the basics.
5. Leverage the experience of MathWorks experts.

## Experiment with a Small Piece of the Project

A recommended approach is to take a new area of the system, build a model of the software behavior, and generate code from it. This is a small change and one that a team member can make with a minimal investment in learning. This will demonstrate that:

- A model can be simulated to work out the bugs in the algorithms much more simply and with greater insights than dynamically testing C code on the desktop.
- High-quality code is created without hand coding.
- The code matches the behavior of the model.

"With Model-Based Design we spent most of our time developing and optimizing the system. Almost no time was spent implementing it in C or debugging code. There's no difference in performance between the Simulink model running on a laptop and production code."

— Karel Viaene, CNH

## Build on Initial Modeling Success

The next valuable activity in adopting Model-Based Design is system-level simulation. Typically, once a segment of code has been shown to work well, the developer will incorporate it within the rest of the application and discover hardware-software integration issues. (These are the same issues one would find if all the code is hand written.)

By contrast, system-level simulation enables the engineer to perform system integration virtually. To do this, an engineer needs to create an adequate model of the hardware in the system and the basics of the environment around the system. Model fidelity should be high enough to ensure interfacing signals have the right units and are connected to the right channels, and the right level of dynamic behavior is captured to test the algorithm.

The results offer an early view of how the hardware and the embedded software will behave. This is especially valuable if the hardware is in development and does not yet exist or is expensive to prototype.

"Typically, significant modifications to the prototype are needed during development. For this project, the performance of the actual hardware matched the simulation results from our Simulink model of the maglev system, so we did not need to modify our test system. Eliminating multiple prototypes saves time and when the prototype costs \$20,000 to \$30,000, as it did for this project reduces costs significantly."

— Cheol Hoon Park, Korea Institute of Machinery and Materials

## Use Models to Solve Specific Design Problems

Engineers can gain targeted benefits even without developing full-scale models of the plant, environment, and algorithm. For example, suppose a team needs to decide on the required parameters for a solenoid used for actuation. They can develop a simple model that draws a conceptual “control volume” around the solenoid, including what’s driving it and what it’s acting on. The team can test various extreme operating conditions, and derive the basic parameters without having to derive the equations. Through simulation, engineers develop the forces needed and then feed them back to the hardware design activity. This model can then be stored for later use to solve a different design problem or for use in the next product development project.

“The simulations that we run with Simulink and Simscape let us evaluate various design concepts and parameter values, reducing the risk of redesign later in the development cycle. Once we identify an optimal design via simulation, we can build it. We anticipate a 30–50% reduction in prototype builds as a result.”

— *Jae Yong Lee, Volvo Construction Equipment*

## Stick with the Basics

For small teams, the immediate benefits of Model-Based Design include the ability to:

- Create component and system models
- Use simulations to validate and test designs early and continuously throughout development
- Generate C code automatically for prototyping and testing

Later, teams can consider advanced tools and practices that include modeling guidelines, requirements linking, compliance checking, and production code generation.

Additionally, they can adopt and adhere to Agile development practices by connecting with continuous integration (CI) systems to automatically test and verify models and code. They can also extend use of system models to the systems in operation, to enable applications such as predictive maintenance and digital twins.

However, for small teams, these can be addressed at a later time.

“A small team of engineers pulled together an autonomous vehicle with off-the-shelf hardware and control algorithms developed and implemented with Model-Based Design. Though the system isn’t production-ready, it does demonstrate important design concepts with a pragmatic design approach.”

— *Dr. Mark Tucker, TMETC*

## Leverage the Experience of MathWorks Experts

MathWorks engineers have worked closely with large and small work groups globally. They have years of experience helping small teams apply Model-Based Design to reach an end goal. A good first step for a small team interested in adopting Model-Based Design would be to work with MathWorks to help your team come up to speed quickly.

“MathWorks Consulting Services helped us model our signal and image processing pipeline in Simulink, run simulations to verify that it was producing quality images, and then implement the design on an FPGA—all while ensuring that we acquired the know-how to do the work ourselves the next time.”

— John Beck, Infraredx

## Success with Model-Based Design

By adopting Model-Based Design incrementally, small engineering teams in organizations of all sizes have consistently achieved immediate and tangible results:

- Faster time to first demonstration
- Faster time to market with a high-quality product
- Expanded capacity for developing complex systems

Small teams depend on the efforts of engineers who put in long hours, show high dedication, and move quickly. With an incremental approach, these teams can smoothly adopt Model-Based Design and perform at even higher levels of speed, competence, and design quality.

## Learn More

- [Model-Based Design](#)
- [Simulink](#)