

FAQ: Adopting Model-Based Design

Q. Is there a recommended way to adopt Model-Based Design?

A. MathWorks has helped many organizations successfully adopt Model-Based Design. View the adoption roadmap that serves as a starting point for customers. Once MathWorks understands your organization's specific challenges, goals, and vision, we can work with you to develop a custom adoption roadmap.

Q. How do we choose a project that is right for piloting the adoption of Model-Based Design?

A. There are several best practices for selecting the right project including:

- Picking a project with high level of algorithmic complexity - one in which the traditional development process might cause delays or other problems. Ideally, the project can be divided into stages that correspond to the adoption road map.
- Choosing a project that is not on the critical path. If you must choose a critical project, you'll need to create a backup plan.
- Selecting a project that can be represented in the available modeling domains (MATLAB, Simulink, Stateflow, and Simscape). Components with close ties to hardware, such as device drivers or operating systems, are better suited for lower-level languages such as C code.

Q. How do we select the right team to work on the first project?

A. The skillset required to adopt Model-Based Design is similar to skillset needed for embedded controls systems and software development. The key considerations for selecting the team are as follows:

- Size: The team should consist of 1 to 3 engineers who are knowledgeable in systems, controls, software, and testing.
- Availability: One of the team members should work on the project full time and preferably be responsible for modeling and algorithm development.
- Attitude: Team members should be open to change and improving the status quo. They should be collaborative consensus builders and early adopters.
- Experience: Team members must have thorough knowledge of the work area and current development life cycle. Interns, new college grads, and those with a narrow knowledge set are not recommended.

Q. How do we quickly come up to speed?

A. MathWorks offers a variety of resources to help customers ramp-up quickly with Model-Based Design. Customers can take advantage of free webinars, workshops, best practice papers, technical support, training services, Application Engineering/Pilot Engineering support, and consulting services.

Q. What happens to existing code when we move to Model-Based Design?

A. Model-Based Design provides facilities to integrate legacy code that supports both simulation and code generation. The system architecture model can thus contain both intrinsically modeled and legacy components. This mixed approach allows for a phased migration of legacy components while supporting system simulation, verification, and code generation along the way.

A review of the existing code base should be performed to prioritize the migration of each component. Priority should be given to migrating components with the following attributes:

- Frequently changed components or those that may need to be changed in the future
- Components with quality issues
- Complex and difficult to maintain components
- Those that are clearly represented in the modeling domain

Q. How will adopting Model-Based Design impact my organization?

A. Adopting Model-Based Design is an evolutionary change, as opposed to a revolutionary change. It builds upon your existing development process. More time is devoted to front end design, which results in less time spent on back end testing.

The impact of adopting Model-Based Design varies depending on your objectives and the current state of your organization, especially when considering:

- Staff skillset and expertise
- Resource availability
- Development process
- Tools and infrastructure

Impacts on the organization include:

- Integrating new tools into existing processes
- Automating tasks that are currently done manually

MathWorks has resources to help your organization adopt Model-Based Design. Once the details of your organization are understood, MathWorks can help identify your specific risks and help develop a risk mitigation strategy. As with introduction of any new technology, there is upfront investment in people, processes, and tools. You can manage this impact and investment using a phased approach.

Q. How are engineering roles affected by the introduction of Model-Based Design?

A. Model-Based Design does not replace engineering expertise such as control design and software architecture. With Model-Based Design, control engineers' roles expand from providing paper (non-executable) requirements to providing proven (executable) requirements and designs in the form of models and code. Software engineers continue to be critical, although less time is spent hand-coding application software. This leaves more time to model architecture, hand-code platform (OS, device drivers, etc.) software, and perform system integration.

Due to the model-centric nature of development, algorithm and software engineers influence the system-level design earlier in the development process. This results in early detection and removal of requirements and design errors, and it leads to fewer implementation errors.

Q. What organizational pitfalls should we consider?

A. One of the biggest challenges to successfully adopting Model-Based Design is in the organizational inertia to resist process changes. There are two types of failures: The first is the failure to adopt, and the second is the failure to realize the full benefits of Model-Based Design.

These failures often occur for the following reasons:

- Underfunding a project. All significant process improvements require dedicated people and time.
- Going it alone. MathWorks has years of experience – organizations that have successfully adopted Model-Based Design have often taken advantage of this.
- Lack of a clearly defined goal. It is necessary to take the time to define the objective.
- Failure to adopt metrics and milestones. You can't improve what you don't measure.
- No committed champions. Volunteers at both the technical and management level are better than conscripts. Motivation to succeed is half the battle.