Massachusetts Institute of Technology

Models in Engineering
Key Takeaways

WEEK 2: MAKING A MODEL

Qualities of Great Models: Weeks 1-2

For the duration of this course, we will be building up a list of the "qualities of great models" with the purpose of distilling guidelines for building and using models effectively. Every week the list will grow with new items presented in the course.

From what you have learned in Weeks 1 and 2, there are seven additional qualities (Qualities 4-10) that should be taken into consideration when developing a model:

1. **Model Fidelity**: The model should have the appropriate level of fidelity relative to the decision under consideration and the phase of the design. More fidelity than necessary can make it difficult to evaluate the model and can waste resources which could have been utilized elsewhere. Less fidelity than necessary can result in poor decisions or overconfidence. The choice of fidelity depends upon the system requirements and operating conditions.

2. **Model Credibility**: Credibility refers to whether the results of the model are believed by decision makers. If models are not credible, decision makers will not make decisions based on the model's output, which might waste precious time and resources, or worse still, they may make incorrect decisions that could put the project at risk. There are many ways in which to build model credibility -- one of which is having a set of standards and processes that guide the evaluation and validation of the model.

3. **Linked to Decision Support**: The real value of models lies in using them in the decision-making process, particularly where we have to trade-off between a lot of parameters. The best models make evident how their outputs are to be used in decision-making.

4. **Understandable and Well–Organized**: More than just documented, is it evident where and how you would add to the model? A Model Development Process can lead you to a modularization and organization of the model that is more understandable.

5. **Well Formed for Optimization**: Has the model been constructed to facilitate optimization if necessary? Does it expose relevant optimization information, like gradients or convexity?

6. **Complete Relative to Scope and Intended Purpose**: The model does not leave out any important physics or dynamics. For example, a model of the economy that includes stock prices but excludes bond prices would be incomplete.

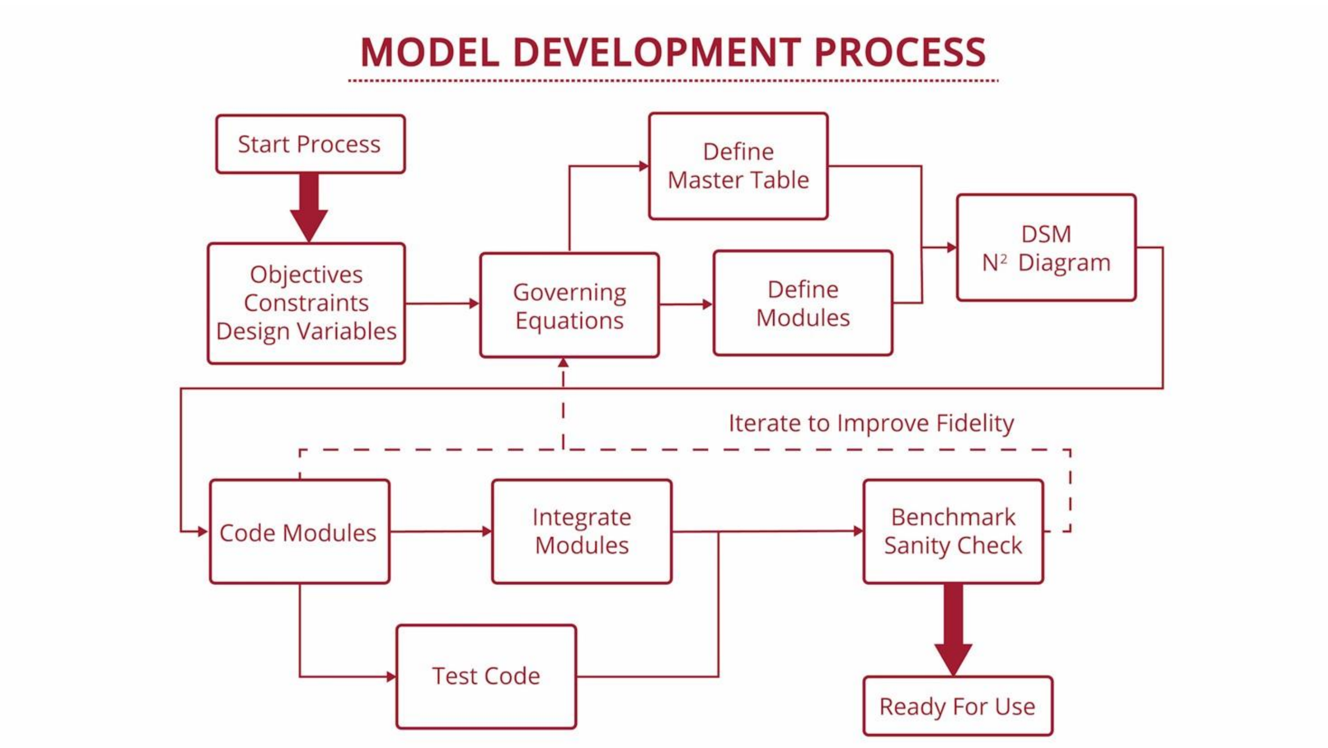
7. **Clear Scope**: Particularly for complex systems, modeling the entire system may not be needed for the project. The breadth must be agreed upon at the beginning of the effort: Which system or part thereof will be modeled. For example, in the automobile industry, the initial design iteration may model all of the components of an engine (broad modeling), while other efforts may work with stable interfaces and model only one component (narrow modeling).

8. **Internally Consistent**: The model does not contain any direct contradictions. For example, assumptions on the gravity constant in one section are the same in all sections of the model.

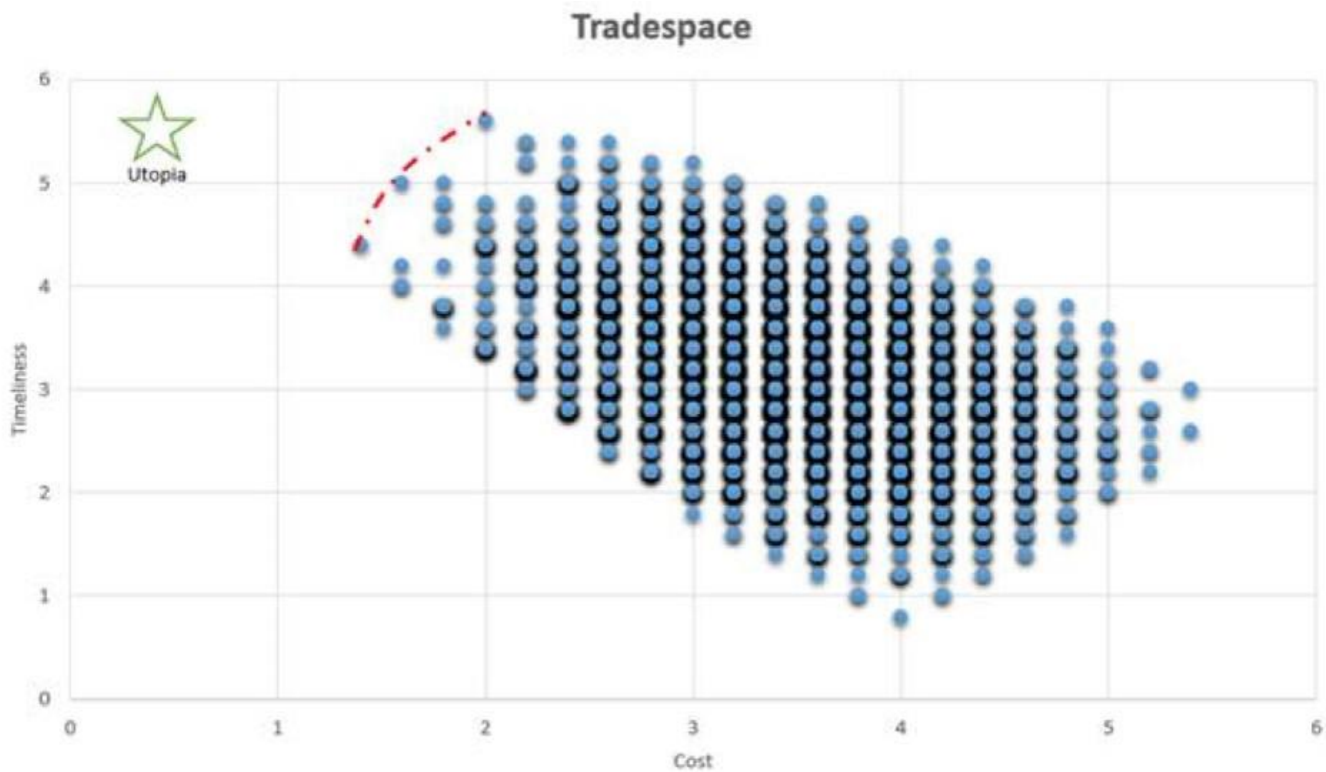
9. **Elegant**: Is the model built using an economy of description -- as simple as possible but no simpler? For example, a model that pulled data from a table and calculated the same parameters from the same data repeatedly might be considered inelegant -- could the result of the computation be stored then reused?

10. **Analyzable and Traceable**: Great models are easily queried. Further, they make it easy to identify what variables or sections of the model contributed to the answer. For example, if the computed thickness of a beam is 6mm, which load cases made the greatest demands on the thickness?

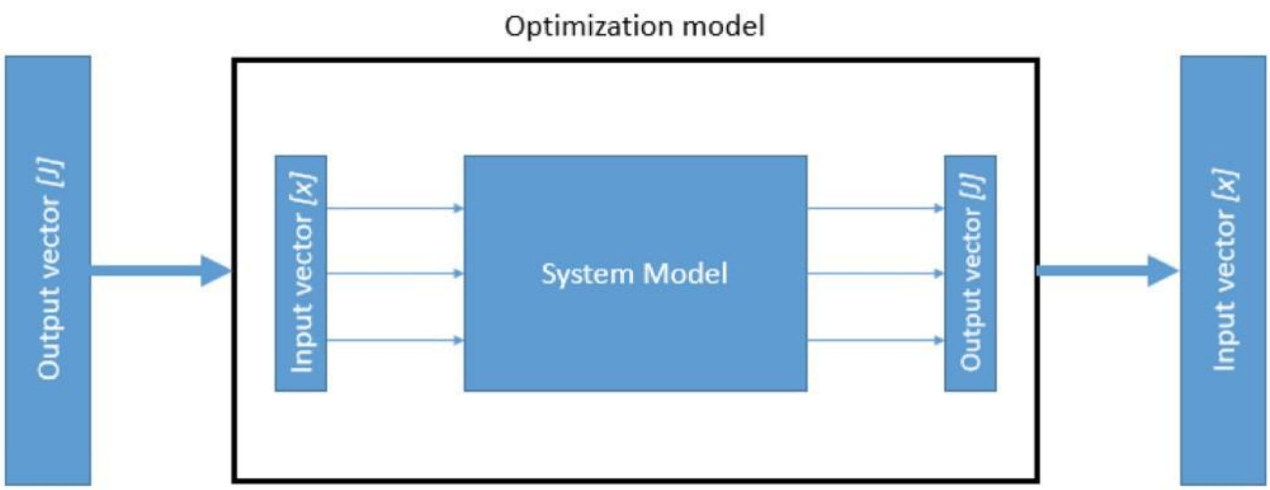
Week 2>How to Make a Model>MDP>

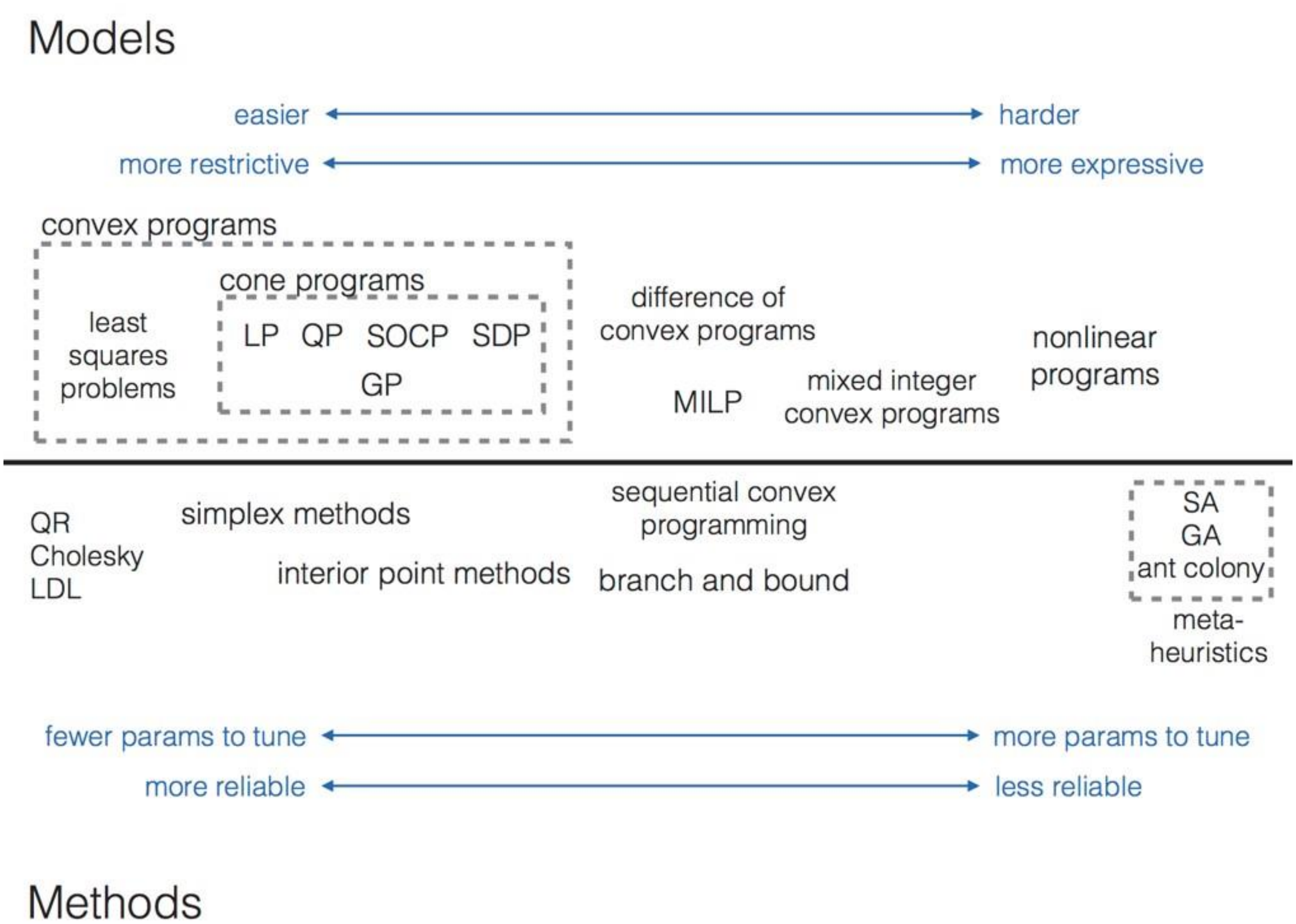


Week 2>Optimizing Models>Tradespace of Model Outputs>



Week 2>Optimizing Models>Optimization as a Model>




Models
easier
harder
more restrictive
more expressive
convex programs
cone programs
least squares problems
LP QP SOCP SDP
GP
difference of convex programs
MILP
mixed integer convex programs
nonlinear programs
QR
Cholesky
LDL
simplex methods
interior point methods
sequential convex programming
branch and bound
SA
GA
ant colony
meta-heuristics
fewer params to tune
more params to tune
more reliable
less reliable
Methods

