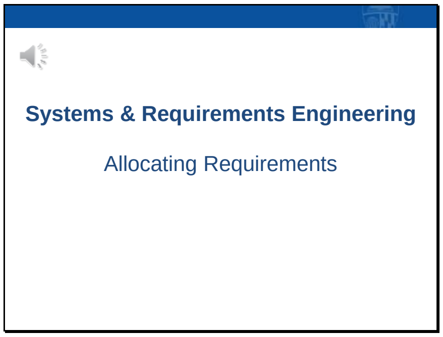
1

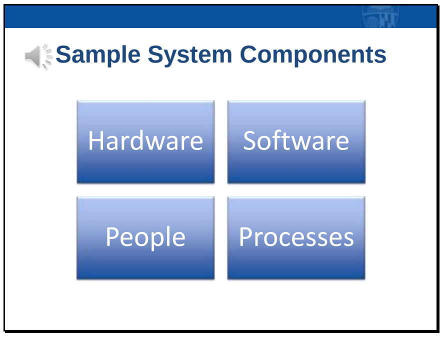


Systems & Requirements Engineering

Allocating Requirements

In this lecture we'll discuss allocating required system functionality to different system components.

2



Sample System Components

Hardware

Software

People

Processes

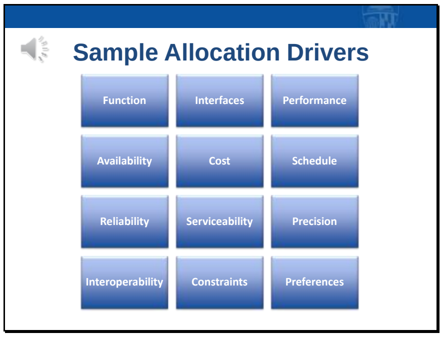
Before we allocate requirements to system components it is important to identify the various subsystems and components of the system. For our purposes, we'll focus on systems that contain hardware, software people and processes.

Hardware can consist of one or more computers, peripheral devices, storage devices, and perhaps special devices such as sensors or measuring devices that could be responsible for sensing changes in environmental variables or reporting measurements. Some applications, like embedded applications, may also require firmware to perform certain functions.

Software could also be decomposed, if necessary, into components such as application software, operating system software, and databases.

Processes could also be responsible for performing needed functionality, either by modifying existing processes or developing new business processes.

3



Sample Allocation Drivers

Function
Interfaces
Performance
Availability
Cost
Schedule
Reliability
Serviceability
Precision
Interoperability
Constraints
Preferences

Once the subsystems and components have been identified, system functions must be allocated to them. In systems engineering this is called functional allocation or functional analysis.

How various functions are allocated is dependent upon each system and its associated parameters, and is typically an exercise in trade-off analysis. Things called drivers, that focus on particular system characteristics, are often used to help make allocation decisions and tradeoffs among the drivers. A dozen commonly used drivers are illustrated here...and I'll elaborate on a few of them.

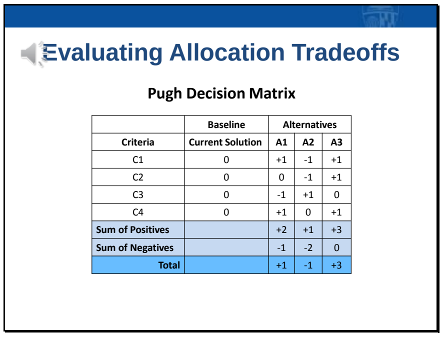
Sometimes the functionality of the subsystems and components make them clear candidates for allocating the overall system functions. In situations where constant monitoring or measurements are required, for example, or where a high level of precision is required, are typically good candidates for allocating to hardware components.

Functionality that requires interfacing to humans, software, or hardware are also considered during allocation. Hardware components typically dictate the interface specifics...either for software interfaces or human interfaces; human interfaces, particularly those that may need to be changed over time, are typically allocated to software.

Physical constraints are another driver. In some projects there may be weight, space, or temperature constraints.

Cost and schedule tradeoffs are obvious drivers that are usually applicable on most every project.

4



Evaluating Allocation Tradeoffs

Pugh Decision Matrix

	Baseline	Alternatives		
Criteria	Current Solution	A1	A2	A3
C1	0	+1	-1	+1
C2	0	0	-1	+1
C3	0	-1	+1	0
C4	0	+1	0	+1
Sum of Positives		+2	+1	+3
Sum of Negatives		-1	-2	0
Total		+1	-1	+3

There are a number of techniques that can be used to help make allocation tradeoff decisions. Most of these techniques use a type of analysis called multi-criteria decision analysis (MCDA). Multi-criteria decision analysis helps us to make decisions about alternatives based on multiple criteria, in which some of the criteria may conflict with one another. Multi-criteria decision analysis is used in a number of engineering disciplines, including systems engineering.

Most multi-criteria decision analysis techniques use some kind of matrix...called a decision matrix. A commonly used form of multi-criteria decision analysis uses a decision matrix called the Pugh Matrix...named after Stuart Pugh, who was a professor at a university in Scotland.

A basic Pugh Matrix is illustrated here...and there are numerous variations. A Pugh Matrix is used to evaluate a number of alternatives compared to some baseline. The baseline is typically an existing product or solution.

The rows of the matrix list the criteria that the decision will be based on. The criteria will be chosen based upon what‘s important in a specifc problem. The columns of the matrix contain the alternatives that will be evaluated. There‘s also a column for the current solution.

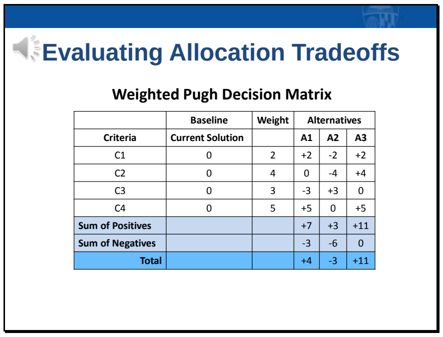
To illustrate the technique, let‘s take an example of replacing an existing system with one of three alternate systems. Then, we‘ll see how we can use the techique to allocate system requirements. So...in this example, we‘re going to evaluate the three alternatives with respect to four decision criteria that we think are the most important. For this example, let‘s assume the criteria are cost, schedule, performance, and reliability.

Our baseline is the current system, so we give it scores of zero. Then, we assess alternative 1 with respect to criteria 1. Is it better, the same, or worse than the baseline? If it‘s better, we give it a score of +1, if it‘s the same we give it a score of 0, and if it‘s worse we give it a score of -1. For our example, let‘s assume it‘s better...so we give it a +1. We then assess and score the remaining three criteria. Let‘s assume alternative one is the same for the second criteria, worse for the third criteria, and better for the fourth criteria.

We would then assess and score the second and third alternatives in a similar manner.

Then...we add up the scores in each column to get an overall total. In this case, the best choice is alternative three, with a total score of plus 3. It‘s also common to sum the positives, negatives, and neutrals, so that more insight into the pros and cons of each alternative can be visualized. This technique is useful because it helps us to make more objective and rational decisions.

5



Evaluating Allocation Tradeoffs

Weighted Pugh Decision Matrix

	Baseline	Weight	Alternatives		
Criteria	Current Solution		A1	A2	A3
C1	0	2	+2	-2	+2
C2	0	4	0	-4	+4
C3	0	3	-3	+3	0
C4	0	5	+5	0	+5
Sum of Positives			+7	+3	+11
Sum of Negatives			-3	-6	0
Total			+4	-3	+11

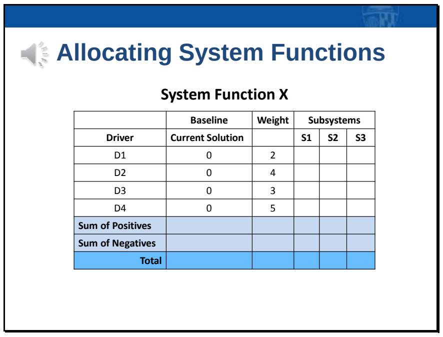
There are several variations on the scoring method. One variation is to weight the criteria according to importance and then multiply the alternatives by the weights.

In this example will use a set of weights that range from 1 to 5. let‘s assume that criteria one is given an importance rate of two, and that criteria two is considered twice as important...so we‘ll give it a weight of 4. Criteria three is somewhere between the first two criteria, so it gets a weight of 3. And, criteria four is the most important, so it gets a weight of 5.

Then, the +1, 0, and -1 scores for each of the alternatives are multiplied by the corresponding weights...so...the scores for the first criteria are multiplied by 2, the scores for the second criteria are multiplied by 4, and so forth.

Yet a third scoring variation is to expand the +1, 0, -1 scoring system to a two point system...where +2 would correspond to much better than the baseline, +1 would correspond to better than the baseline, and so forth.

6



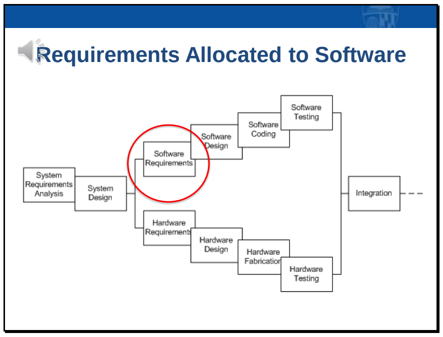
Allocating System Functions

System Function X

	Baseline	Weight	Subsystems		
Driver	Current Solution		S1	S2	S3
D1	0	2			
D2	0	4			
D3	0	3			
D4	0	5			
Sum of Positives					
Sum of Negatives					
Total					

Now...how does one use this decision analysis technique to allocate system functionality to subsystems or components? There‘s a couple of ways this could be done, but the most straightforward way is to take each system function and construct a Pugh Matrix for it. The rows of the matrix correspond to the important drivers, and the columns correspond to the subsystem that the function is allocated to.

7



Requirements Allocated to Software

At this point we‘ve discussed allocation of system level requirements to different system components...and we will assume that the appropriate system requirements have been allocated to software. So, going forward, we will leave the topic of systems engineering and focus on software engineering ...starting with software requirements.