PAPER SAS3165-2019

The Science of Workflow Management: Modeling Business Processes in SAS® Simulation Studio

Dave Manet, SAS Institute Inc.

ABSTRACT

As managers, we are often asked to ensure that our business processes are optimized. There are many software solutions for simply tracking work. But true process optimization can be attained only by carefully examining inputs and studying how varying those factors influences the outcome. Without specialized tools such as SAS® Simulation Studio, obtaining accurate results is very challenging due to the complexity of modern business. In this paper, we use SAS Simulation Studio to conduct experiments of how changes in work volume and resource availability impact process efficiency and capacity of an essential business process. Modeling a process in a simulated environment allows for true A-B testing. The point-and-click interface is easy to use and empowers even novice users to translate real-world scenarios into reliable estimates. The desktop deployment makes it easy to install for businesses of any size, and integration with JMP® enables quick analysis of results, which promotes experimentation and data-driven decision-making. Business process management is often called a science and an art; analytics helps ensure that the science is solid.

INTRODUCTION

Competitive edge is defined as the ability to stay ahead of present or potential competition. It is our responsibility as managers to diligently monitor day-to-day operations and actively pursue opportunities to establish competitive advantage over other companies. One area where competitive advantage can be achieved is operational effectiveness. Anyone can have ideas on how efficiency can be improved. In today's business environment, a gut feeling or a hunch is not always enough. To draft a persuasive strategy, data is needed to back up your claims. This is called data-driven decision-making.

You might be asking yourself "How can I have data until a process change is implemented, measured, and performed over time? I have a pretty good idea of how my business performed last week, last month, or over a few years, but is there a way to use this information to evaluate different ideas before time and resources are expended on strategy that might or might not work?"

Yes, there is a way, and it's called discrete-event simulation. Discrete-event simulation is a simple yet versatile way of modeling the operation of a dynamic system in a virtual environment. Suppose that you are a pilot performing training in a flight simulator. The virtual environment of a flight simulator allows you to try out different scenarios without putting you or your company assets in danger. Simulation is a great way to diagnose complex workflows for issues with utilization, bottlenecks, capacity, and various forms of stress testing.

With the right tool at your disposal, your ideas, scenarios, and years of operation can be simulated and analyzed within minutes, enabling you to make a better decision more quickly. The right tools for the job are SAS Simulation Studio and JMP.

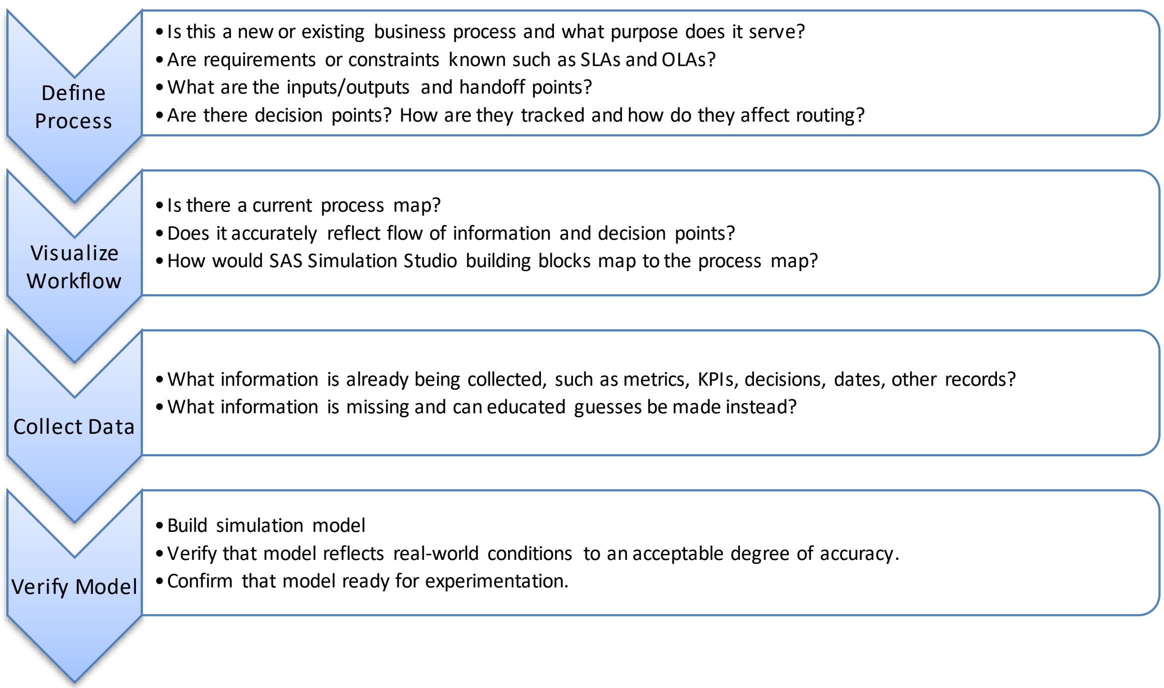
SAS Simulation Studio is a powerful GUI tool used to model behavior of a real-world system in a simulated environment. SAS Simulation Studio is designed to interact with both SAS® software and JMP software so that you can conduct sophisticated statistical analyses of your results. Data generated by the model can be saved as a SAS data set or JMP table for later analysis. You can also use a SAS block included in the basic template of modeling blocks to execute SAS or JMP code directly from SAS Simulation Studio.

My goal with this paper is to walk you through the approach that our team took when we wanted to optimize one of our essential business processes. Code samples and supporting materials are made available so that you can follow along or experiment with the model yourself. You can download the sample files from: https://github.com/sascommunities/sas-global-forum-2019/tree/master/3165-2019-Manet

Before we get into conducting experiments, let's take a few minutes to understand the business process we'll be working with and how we translated it into a simulation model.

BUILDING SIMULATION

In this section we will go over the business process and how we went about verifying accuracy of our model. Key steps in our model building process are summarized in the table below.



DEFINE PROCESS

What are we simulating?

We will be simulating a how requests move through a team of Solution Architects over a one-year period. This is a simplified version of our real-world workflow with many unnecessary elements being excluded.

Before a SAS solution is deployed, a Solution Architect translates sales requirements into a technical blueprint for use by the Build team. A technical review occurs before the blueprint is accepted. After the blueprint is accepted, it is handed off to the Build team for implementation. In the meantime, the Architect performs document finalization and customer on-boarding activities. After the request is completed, the Solution Architect is assigned to the next customer, and the process starts over.

On the team there are five Solution Architects. Each Architect can track up to two requests at a time, and occasionally they can be overprovisioned to track up to four requests at a time. Each request is reviewed and scheduled based on how many Solution Architects are available during the week. If work-week capacity is reached, the request is pushed out to next available week. There are also occasional high-priority requests, which are placed ahead of others. If an Architect is off work due to vacation or sick leave, their in-progress workstreams are placed back in the queue and are picked up by the next available Architect ahead of new work, but after high-priority requests.

VISUALIZE WORKFLOW

A visual representation of the architecture review process is outlined in Figure 1. As you can see, the architecture review process has seven phases, with completion of a technical review being a critical milestone at phase five. After the architecture passes technical review, it is handed off to other teams while non-critical onboarding activities are completed.

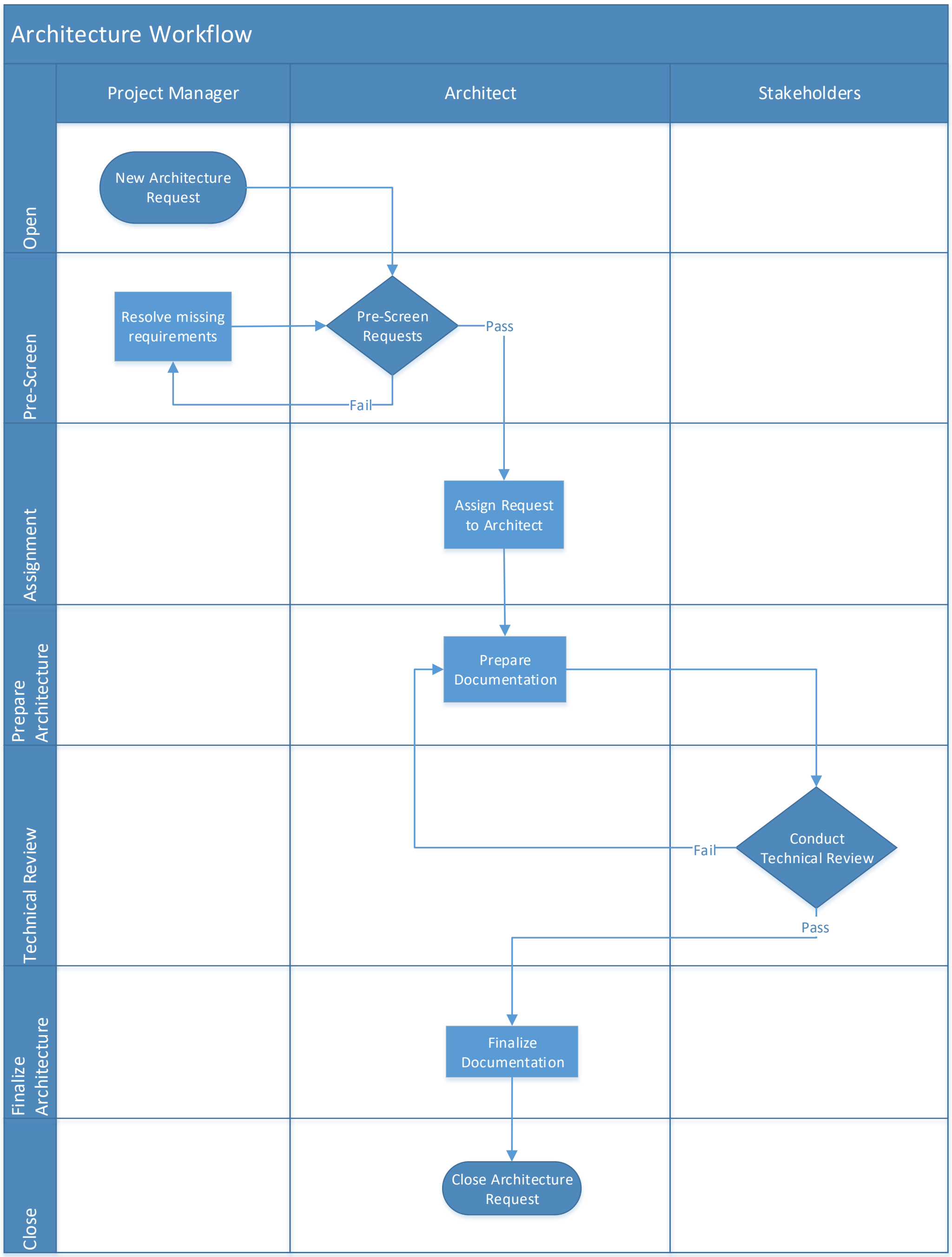


Figure 1. Architecture Review Process

Using the workflow example in Figure 1, we can start identifying high-level requirements for the SAS Simulation Studio model, such as queues, decision points, delay blocks, and sections where conditional probability will need to be used. Figure 2 shows how these components can be implemented.

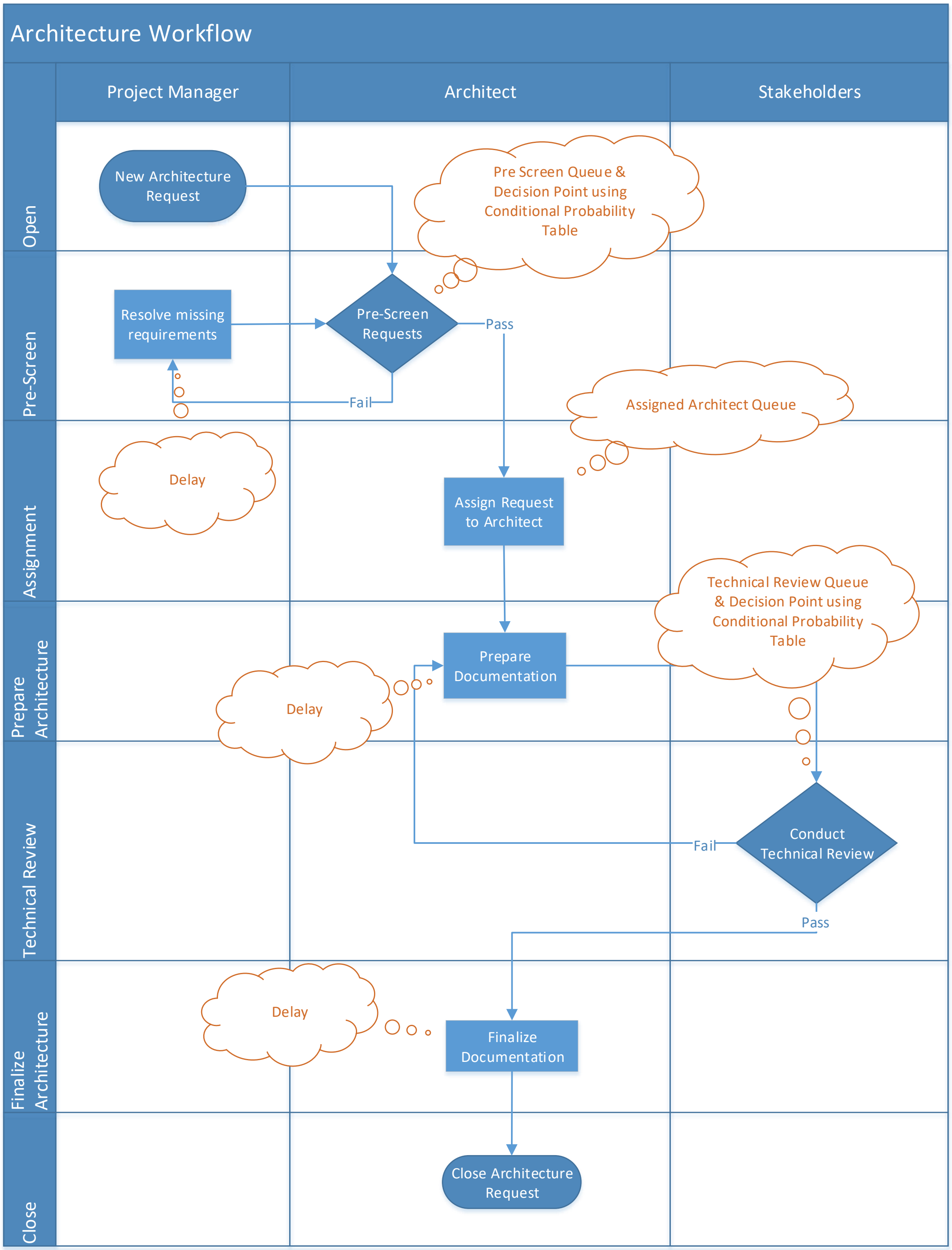


Figure 2. Implementing the Architecture Review

COLLECT DATA

Data for our decision point probabilities, durations, request volume, and so on, will be used to validate the model before experiments can begin. If you are designing a new workflow, this is the time to start thinking about which metrics you'll want to collect upfront. If some data is missing, an educated guess will need to be made. For existing workflows, you should have access to some metrics data before you start. In this paper we will use randomly generated "real world" metrics as our starting point.

Metric	Real World
Requests per year	171
Tech Review Passing Probability	46.7%
Pre-Screening Attempts	1 to 2
Tech Review Attempts	1 to 3
Days in Open State	0 to 2
Days in Pre-Screening State	0 to 1
Days Assigned to Architect State	1 to 3
Days Preparing Documents	3 to 20
Days Finalizing Documents	15 to 25
Days to Tech Review Passing	5 to 25
Days to Request Completion	20 to 50
Schedule Ahead Weeks	0 to 1

The table above summarizes key metrics that will be used for model building and validation.

Verify Simulation Studio Model

Now you are ready to begin the process of building a model in a SAS Simulation Studio. However, in the interest of time, this paper will work with a completed model. Before we can experiment with the model, we need to run a baseline experiment to verify that our model is "accurate enough".

Why is "accurate enough" in quotation marks?

It's because accuracy of a simulation model is subjective to you as a user. You should not expect it to match the real world exactly, because every simulation run is randomized. It is up to you as a user to decide what is "accurate enough". We will do a comparison of results against real-world numbers later in the paper.

Let's go over components of our model.

Simulation Model

If you have access to the sample files, the completed model is stored in the `TechReview` directory. You can download the sample files from: https://github.com/sascommunities/sas-global-forum-2019/tree/master/3165-2019-Manet. A zoomed-out view of the model is shown in Figure 3, and an annotated process map we started with is below the figure. Also, Appendix A contains a zoomed-in view of blocks 1 through 12 in the model.

Here is a brief description of each block and the role it plays in the model:

Block 1 - New Architecture Requests Block – Tracks creation, disposal, and priority of new architecture requests.

Block 2 - Pre-Screening Queue and Delay Block – Tracks the pre-screening queue, seizure of the scheduling resource, and simulation of the delay when the architecture request fails pre-screening.

Block 3 - Pre-Screen Decision Point – Loads conditional probability data set, tracks pre-screening attempts and decisions, and releases a scheduling resource after recording a decision.

Block 4 - Calculate Assigned to Architect Delay – Calculates how many weeks ahead a request will be scheduled, based on request priority, current backlog, and process capacity.

Block 5 - Assigned to Architect Queue – Holding area for Architecture requests, also known as the Architecture Request Backlog.

Block 6 - Preparing Documentation – Using an external data set, the delay duration is calculated. This delay represents how long an architect will work on preparing documentation before attempting to bring it to technical review.

Block 7 - Tech Review Decision Point - Loads conditional probability data set, tracks technical review attempts and decisions.

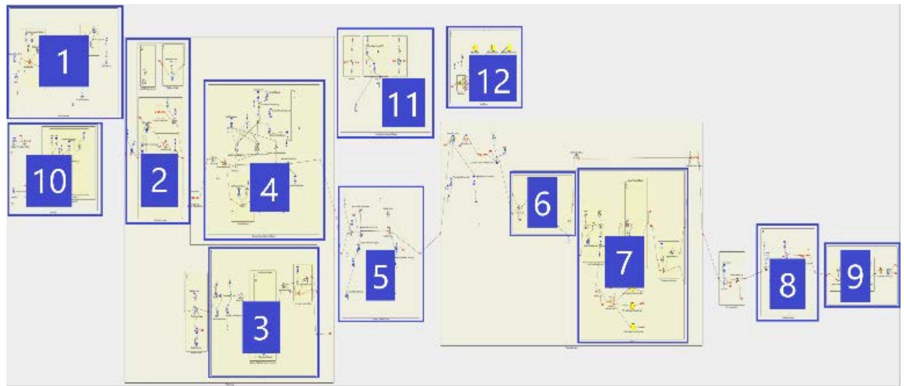
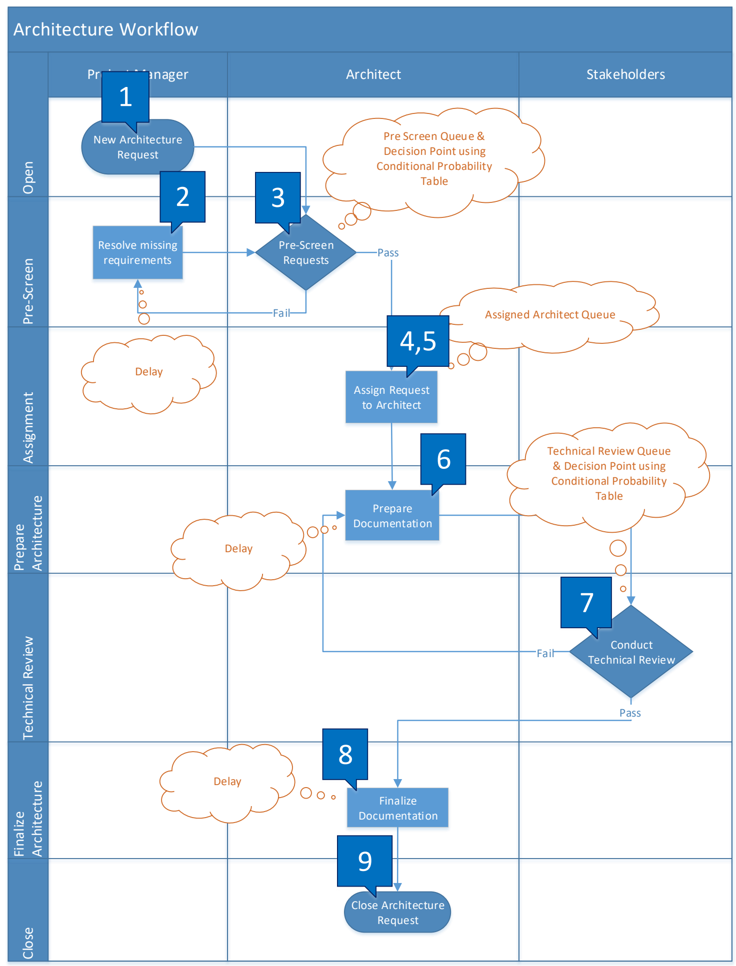


Figure 3. Zoomed-Out View of Sample Model



Block 8 – Finalizing Documentation - Simulates the delay experienced by the architect while on-boarding activities are completed by other teams.

Block 9 – Closing Block – Additional metrics are calculated, and request is sent to the disposer.

Block 10 - Resourcing Block – Supports pre-emption or re-assignment of work when the architect becomes unavailable. Supports seizing and releasing of available architects based on experiment factors.

Block 11 – Seized Architect Tabulator – Tracks the number of currently seized architect resources

Block 12 – Arch Request Stat Tabulator – Calculates durations of architecture request states for every entity

Baseline Experiment

Now we are ready to conduct a baseline experiment that will allow us to verify that our model is "accurate enough".

Inputs

The model we're working with has been configured to support multiple experiment factors. This allows us to make changes in experiment window instead of relying on hardcoded formulas. By anchoring factors this way, you can set up experiments quicker, reuse values as needed and reduce the likelihood of breaking the model due to incorrect block editing.

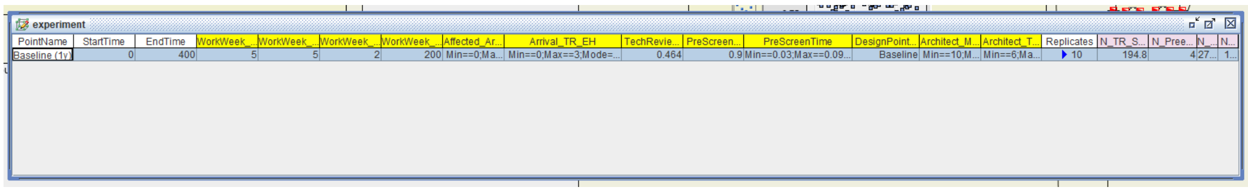


Figure 4. Experiment Design Window

Figure 4 is a screenshot of the experiment design window, where the factors are entered.

For the baseline experiment, the following factors will be used. We chose these values from our understanding of the current process and the real-world metrics we've collected in previous steps.

Factor	Value	Description
StartTime	0	Experiment start time
EndTime	400	Experiment end time. How many days each experiment will run.
WorkWeek_Duration	5	Work week duration Used throughout the model to calculate work cycle, backlog, and process capacity.
WorkWeek_Num_Architect_TR	5	Number of available architects.
WorkWeek_IndividualTRCapacity	2	Workload capacity of an individual architect.
WorkWeek_TROverloadThreshold	200	Permitted overload for an individual architect in percent.
Affected_Architect_TR	Min==0;Max==3; Mode==1.5	Used to calculate how many architects will be impacted by sick leave or vacation.
Architect_MTBF	Min==10;Max==30; Mode==25	Used to calculate when affected architects will be taken out of the resource pool due to sick leave or vacation.
Architect_TTR	Min==6;Max==10; Mode==7	Used to calculate when affected architects will return to the resource pool from sick leave or vacation.
TechReviewPassProbability	0.464	Probability of passing the technical review decision point at each attempt

PreScreenPassProbability	0.9	Probability of passing the prescreening decision point at each attempt.
PreScreenTime	Min==0.03;Max==0.09; Mode==0.05	Used to calculate how long it takes to pre-screen requests.
Arrival_TR_EH	Min==0;Max==3; Mode==1.5	Controls the rate at which new requests are created.
DesignPointLabel	Baseline	Used to copy the design point label in the output data set for the experiment. This makes it easier to analyze results for multiple experiments side by side.
Replicates	10	Number of times each design point should run.

We start the experiment by pressing the Play button.

It takes a few minutes for the simulation to run. The results of your experiment are stored in the \TechReview\results directory. Next we will analyze the results and draw conclusions.

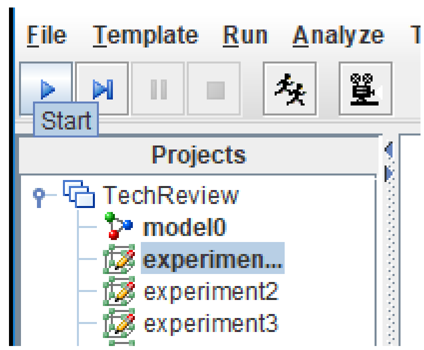


Figure 5. Start the Experiment

Result Analysis

For the purpose of validating the model, the contents of the file \TechReview\results\Close\Bucket_TR\result_tr_0_[0-9].sas7bdat interest us the most.

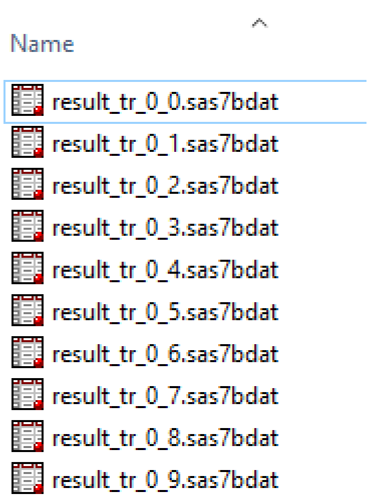


Figure 6. Content of Results Directory

We will systematically go through the generated results using JMP software and record our findings. We will then compare the model output against real-world metrics and draw conclusions.

Calculate Average Completed Architecture Requests Per Year

We will start by calculating average number of architecture requests per year that the simulation model spawned. If the numbers are too far apart compared to the real-world numbers, our model will not pass validation and will have to be adjusted.

We can easily get this information using our model's **N_TR_Serviced** and **N_Carryover** response anchors. If you subtract the values of **N_Carryover** from **N_TR_Serviced,** you know how many technical reviews were completed in a one-year period. If the same formula is applied on the values in the top row, then we will know an average number of completed technical reviews.

In this case, the value for N_TR_Serviced is 194.8, and the value for N_Carryover is 27.2, so 194.8-27.2=167.6, which is rounded up to an average number of completed technical reviews of 168.

Replicates	N_TR_Serviced	N_Preempted	N_Carryover	N_Expedited
10	194.8	4	27.2	18.1
1	192	5	28	14
2	193	5	27	19
3	192	2	27	17
4	198	0	29	23
5	192	0	30	17
6	197	0	20	17
7	201	4	24	18
8	187	1	30	15
9	194	17	26	18
10	202	6	31	23

Figure 7. Experiment Window

Combine Multiple Results Into a Single JMP Table

It will be quicker to analyze data if we combine results into a single table. The process is very simple to execute.

1. Open JMP.
2. Click Import **Multiple Files...** in the File menu.
3. Navigate to the \TechReview\results\Close\Bucket_TR\ directory and wait for a list of results files to appear. After they appear, click Import to create a single JMP table.

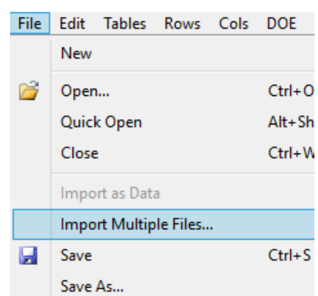


Figure 8. Import Files

4. A single JMP table is now be shown. Now we need to apply data filter to exclude records that have been completed after a one-year cutoff. For example, if a record was created a few days before the year was over, our simulation would still process it. Since we don't need this record for this experiment, we exclude it from our results.

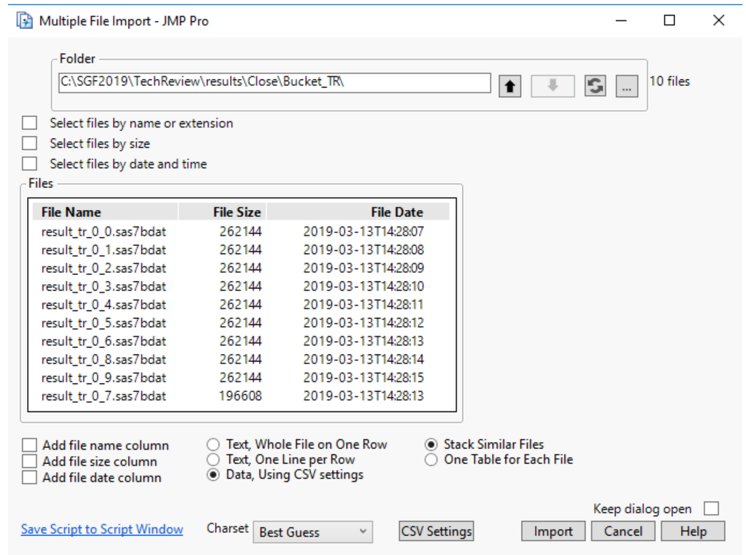


Figure 9. Imported Results Files

5. Click on the data filter icon. Select CompletedAfterCutoff column and then click Add.

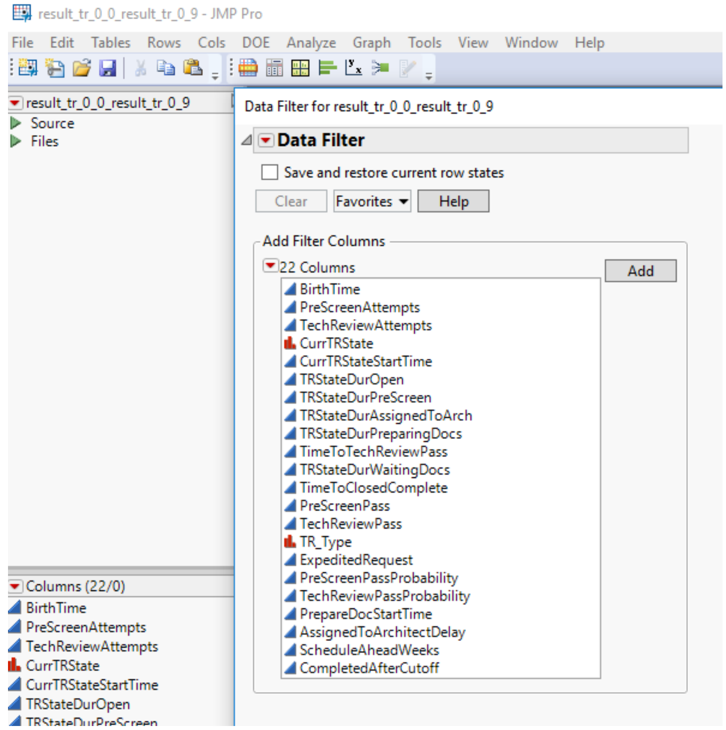


Figure 10. Selecting a Filter

6. Select the check boxes for the Select, Show, and Include options, and drag the CompletedAfterCutoff slider all the way to the left. These selections exclude results that have been completed after the cutoff from our calculations.

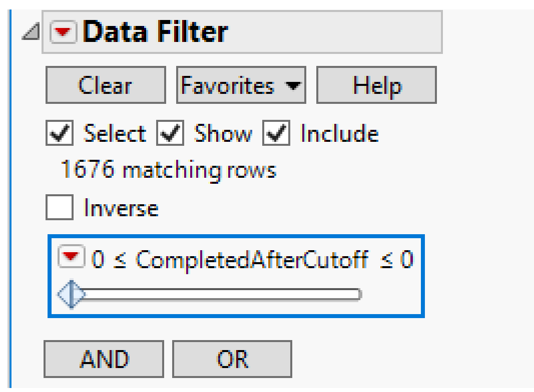


Figure 11. Selecting the Cutoff Options

7. You can verify that the filter was applied correctly in a record list or a row summary.

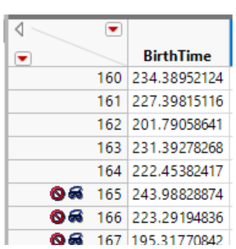


Figure 13. Record List

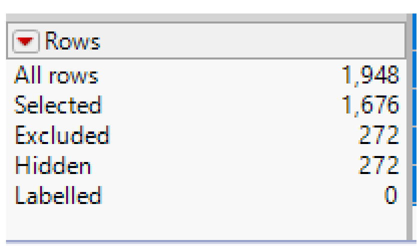


Figure 12. Row Summary

Calculate Technical Review Passing Probability

The technical review passing probability determines the likelihood that an architecture request will pass a technical review decision point. If it fails, the request will be delayed while issues are remediated, and another attempt will be made later. Based on our real-world metrics, we know that a probability of passing technical review is around 46.7%, so we enter this number into our experiment factor. Now we need to confirm that our model is recording decisions with similar probability.

The formula to calculate passing probability or P(Pass) is: $P(Pass) = \frac{Number\ of\ Passed\ Tech\ Reviews}{Number\ of\ Attempts}$

We will pick up from where we left off in the previous step and use the combined results table to calculate this value.

1. Click Summary in the Table menu.

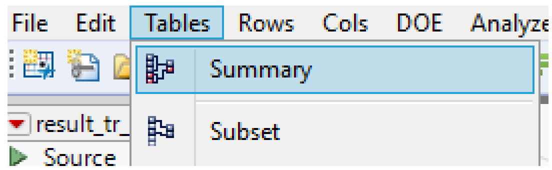


Figure 14. Selecting Summary

2. In the Summary window, select the TechReviewAttempts column. Click the Statistics drop-down menu and select Sum.

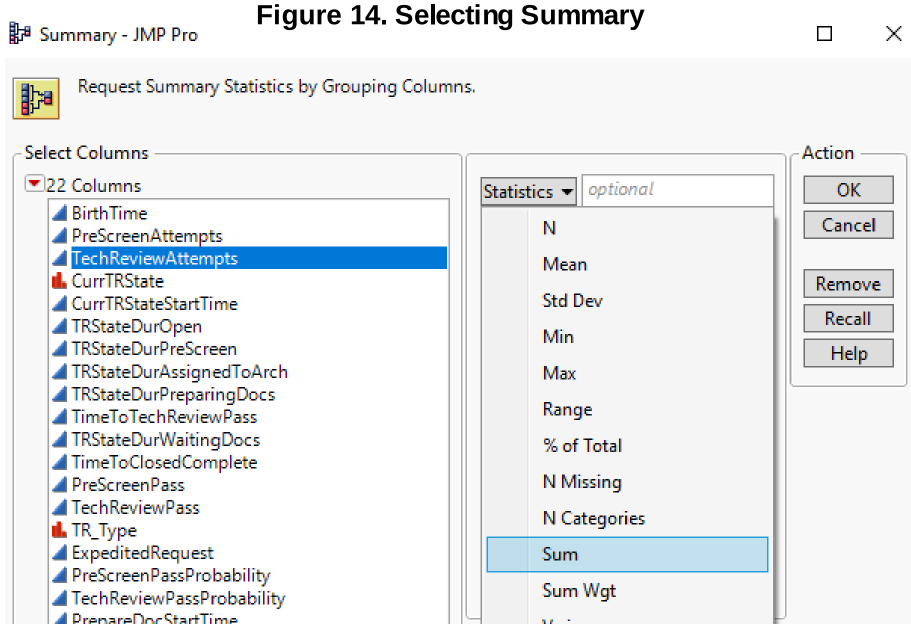


Figure 15. Selecting Sum of TechReviewAttempts

3. Select TechReviewAttempts and click Freq.

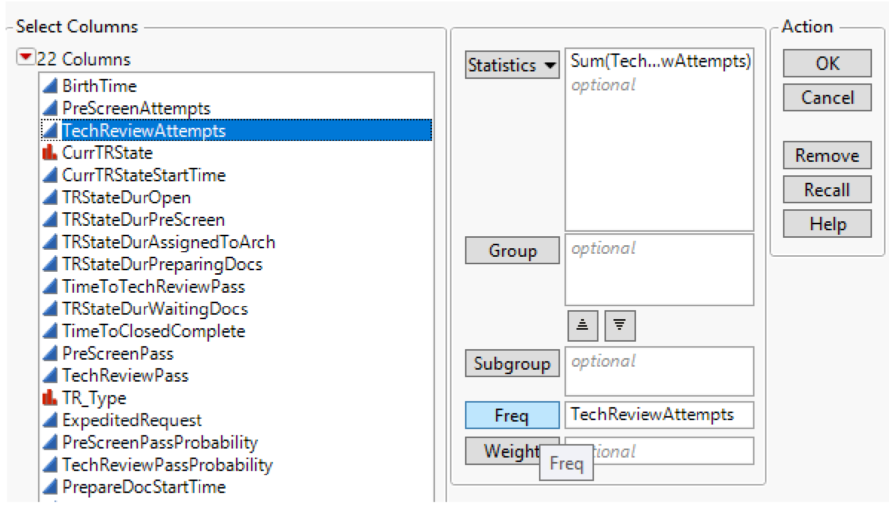


Figure 16. Selecting Freq of TechReviewAttempts

4. A new Summary table appears, containing two columns, N Rows and Sum(TechReviewAttempts). Because only technical reviews that passed appear in the results table, we can use the N Rows column for the value of the *Number of Passed Tech Reviews* variable in our probability equation. The sum of this column gives us the number of attempts that were made before technical review passed, which is our *Number of Attempts* equation variable. Because we have both variables, let's proceed with creating a new column that we'll use for our calculation.
5. Right-click on the empty header to the right of the Sum column, and then click New Columns from the pop-up menu.

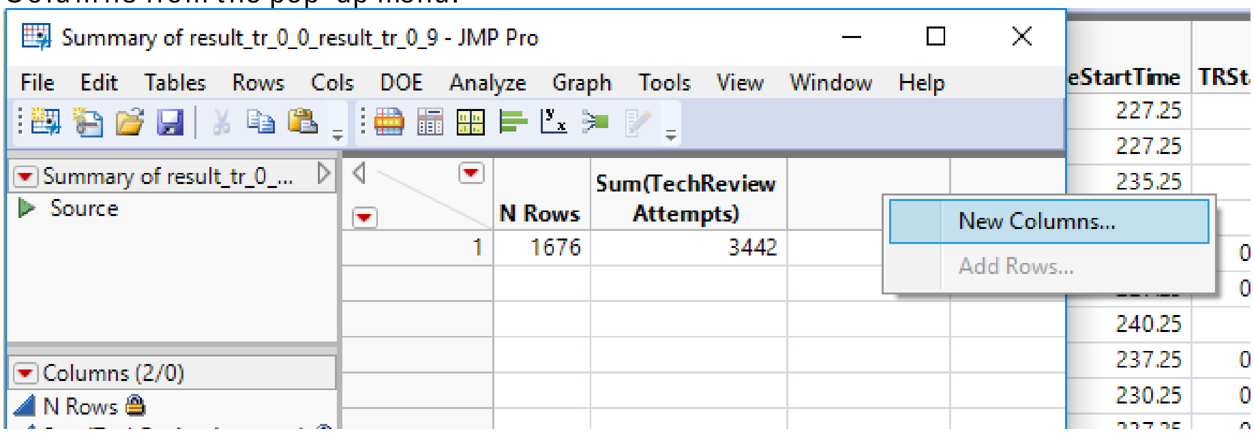


Figure 17. Adding a New Column

6. In the Column Name field, enter P(Pass) and select Numeric in the Data Type field. Click Column Properties to expand a drop-down menu.

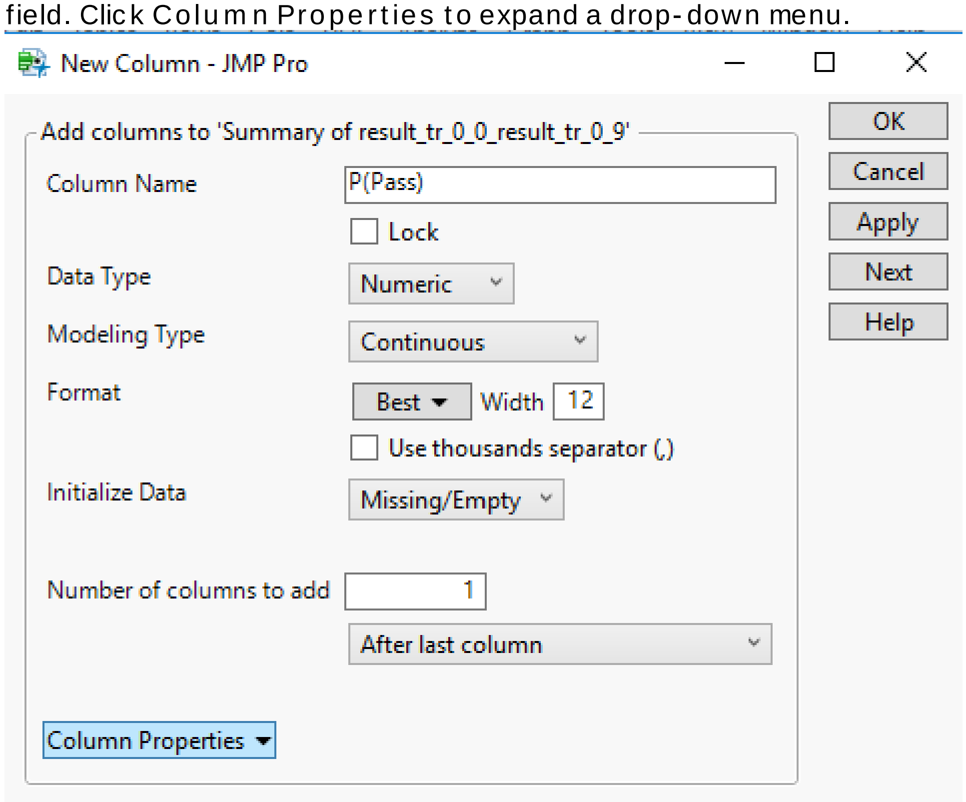


Figure 18. Specifying Column Name and Type

7. Select Formula.

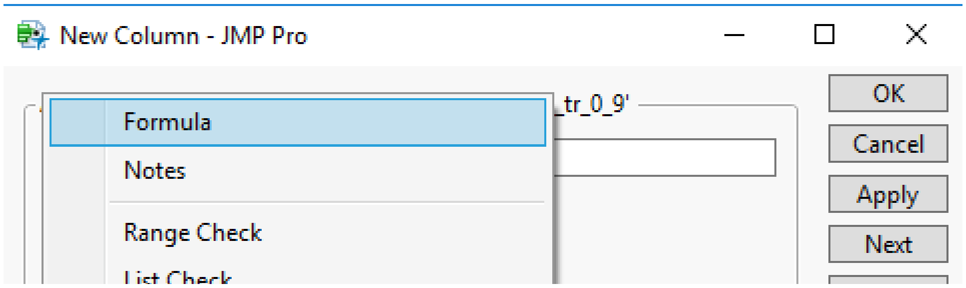


Figure 19. Column Properties

8. Drag and drop columns into a formula canvas so that it looks like the following: N Rows/Sum(TechReviewAttempts). Click OK when you are finished.

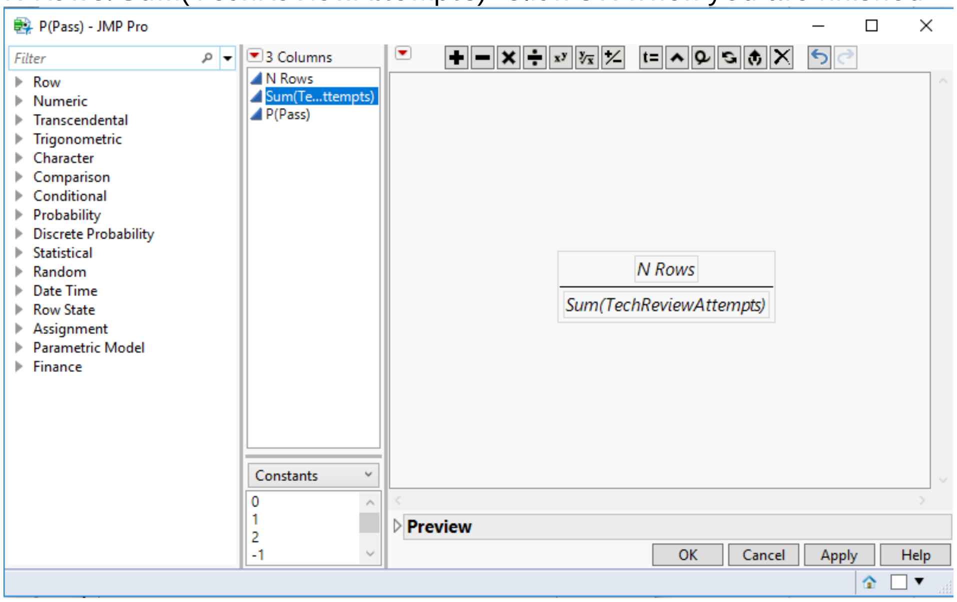


Figure 20. Columns in Formula Canvas

9. The New Column properties window appears. Click the Format drop-down list.

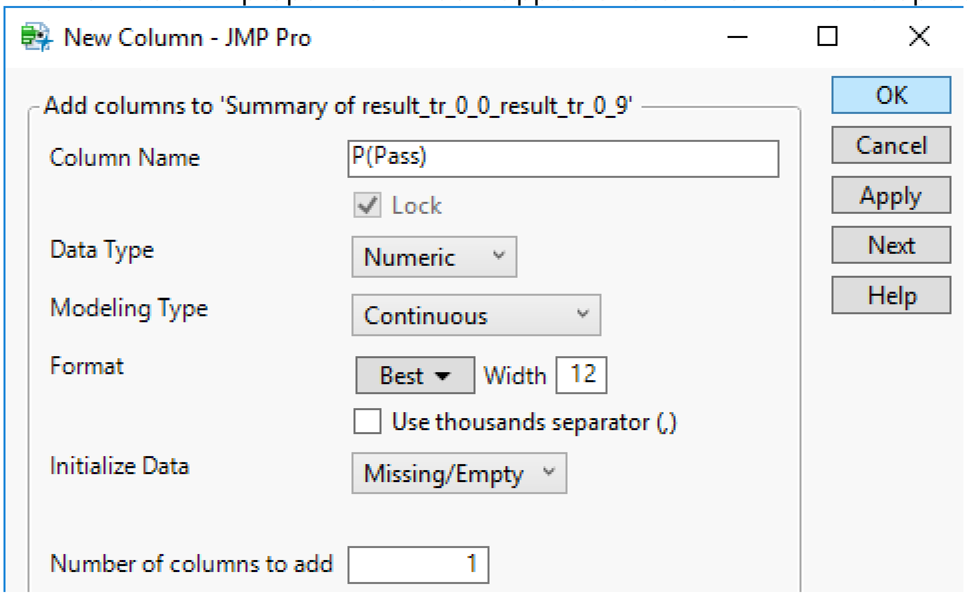


Figure 21. New Column Properties

10. In the Format field, select the Percent, and then specify 1 in the Decimal field.

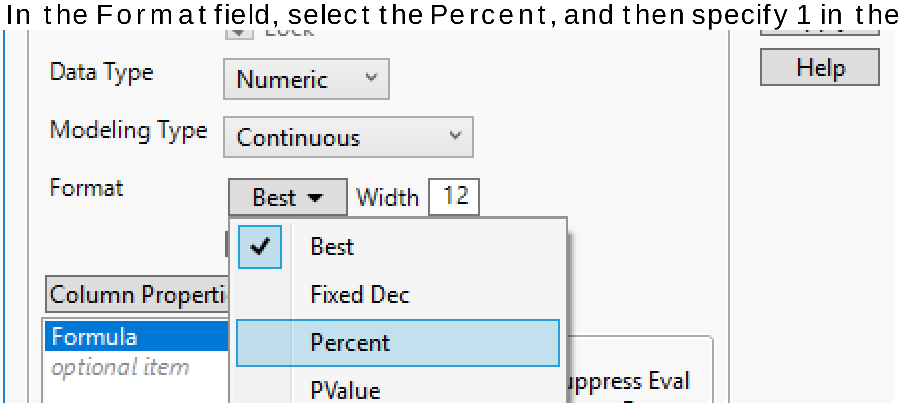


Figure 22. Selecting Percent Format

11. The new column P(Pass) now appears on the Summary table. We record this number in our results list.

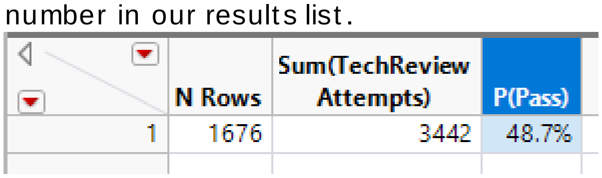


Figure 23. New P(Pass) Column

Calculate Remaining Metrics

The remaining metrics will be calculated all at once using the combined results table that we created earlier.

1. Click the Distribution icon.

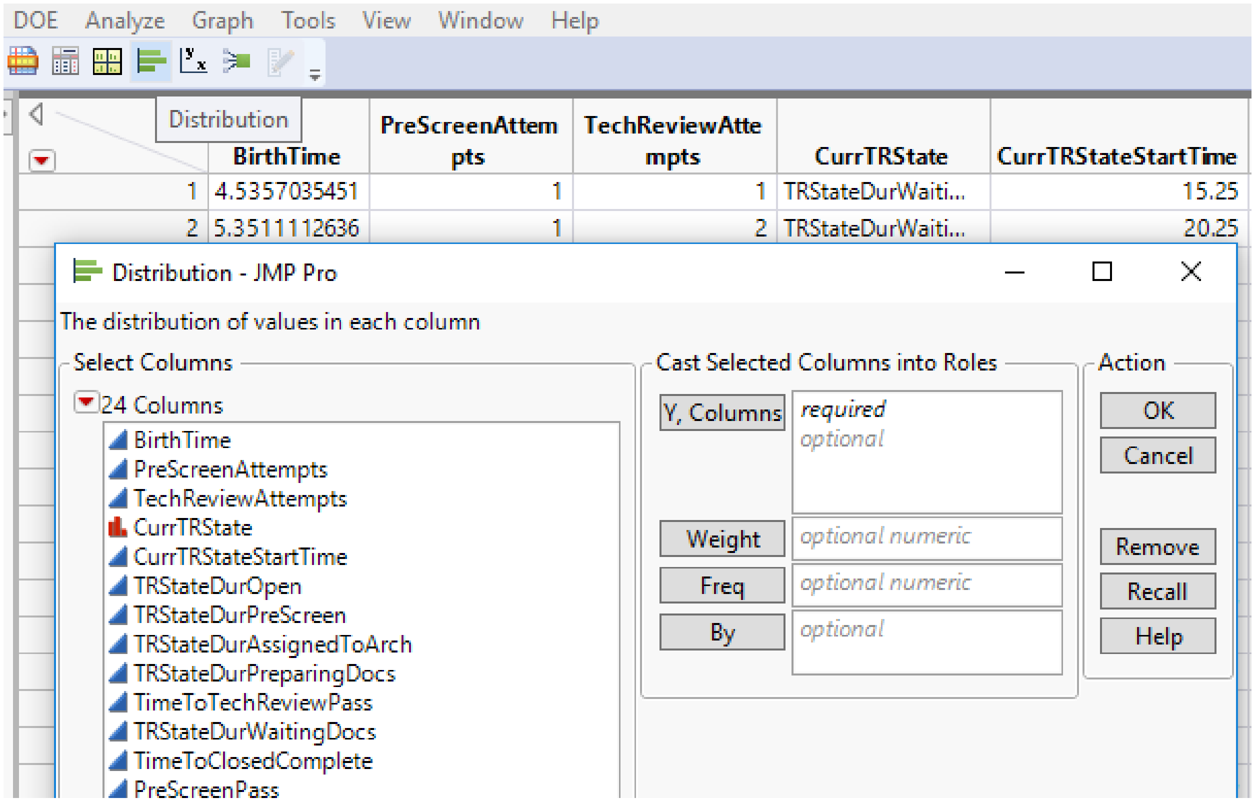


Figure 24. Distribution Designer Window

2. In the Distribution designer window, drag and drop columns into Y, Columns field as shown in Figure 25. After you have selected all Y, Column values, select the DesignPointLabel column and click By. Verify that your screen looks like Figure 25, and then click OK.

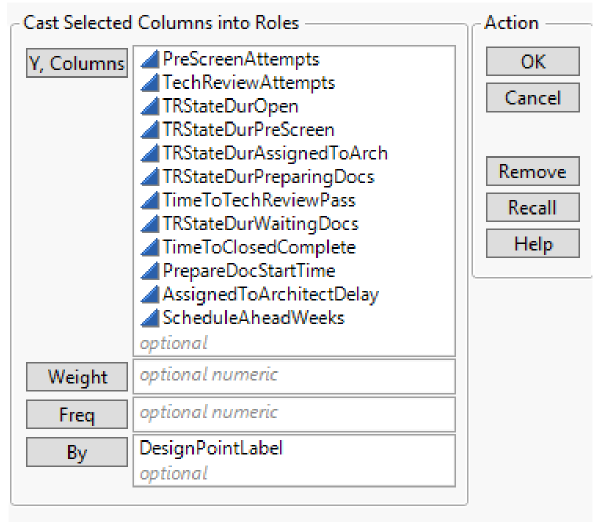


Figure 25. Selecting Columns

3. You will be presented with a lot of information. Of interest to us are highlighted areas, as shown in Figure 26.

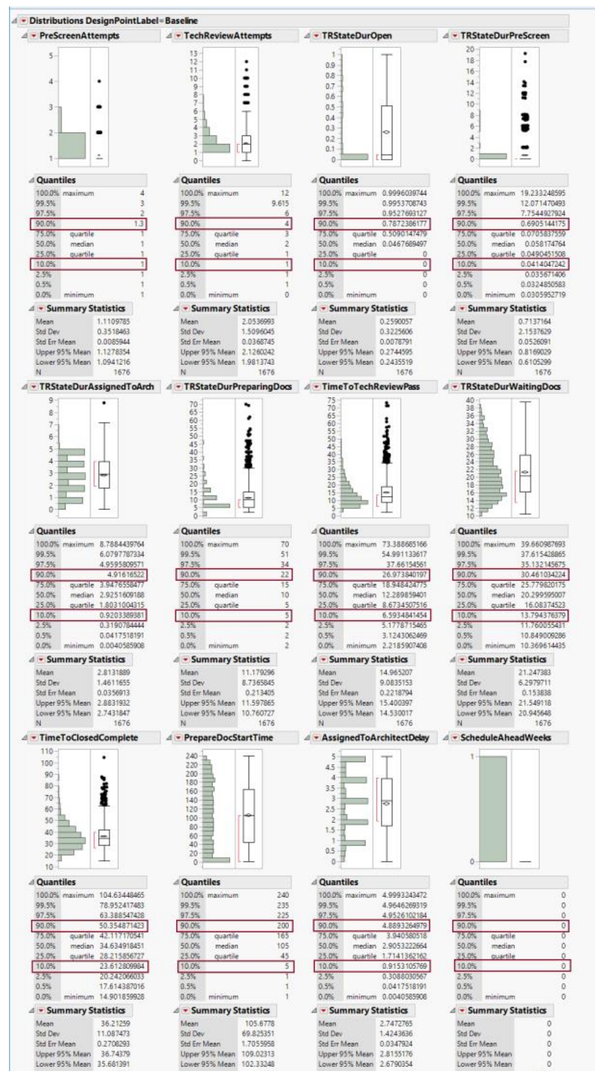


Figure 26. Areas of Interest

Draw Conclusions

The table below shows the results of the baseline experiment conducted in SAS Simulation Studio and our real-world metrics side by side. In my opinion the SAS Simulation Studio responses are very close to the real-world numbers. Based on this comparison, I am able to conclude that our model is "accurate enough" and therefore can be considered valid.

Now that our model was validated, we can begin designing experiments.

Metric	**Real World**	**SAS Simulation Studio**
Requests per year	171	168
Tech Review Passing Probability	46.7%	48.7%
Pre-Screening Attempts	1 to 2	1 to 1.3
Tech Review Attempts	1 to 3	1 to 4
Days in Open State	0 to 2	0 to 1
Days in Pre-Screening State	0 to 1	0 to 1
Days Assigned to Architect State	1 to 3	1 to 5
Days Preparing Documents	3 to 20	5 to 22
Days Finalizing Documents	15 to 25	13 to 30
Days to Tech Review Passing	5 to 25	6 to 26
Days to Request Completion	20 to 50	23 to 50
Schedule Ahead Weeks	0 to 1	0

CONDUCT EXPERIMENTS

Now that we have a validated the simulation model, we can begin experimenting, analyzing results, and drawing conclusions on a wide variety of topics. Key steps in our experimentation process are summarized below, and the process is both straight-forward, and repeatable.

So how can you tie it all of this together and get real-world value out of our model?

Let's suppose that you are a manager and one of your goals is to decrease the amount of time it takes for every request to pass technical review. We know from our current metrics that it takes anywhere from 5 to 25 days on average to reach this milestone. How would you attempt to accomplish this goal? What type of time savings are achievable?

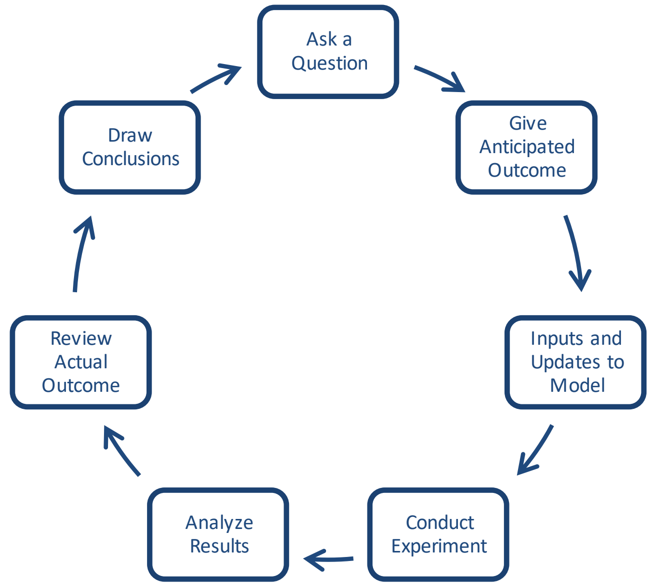
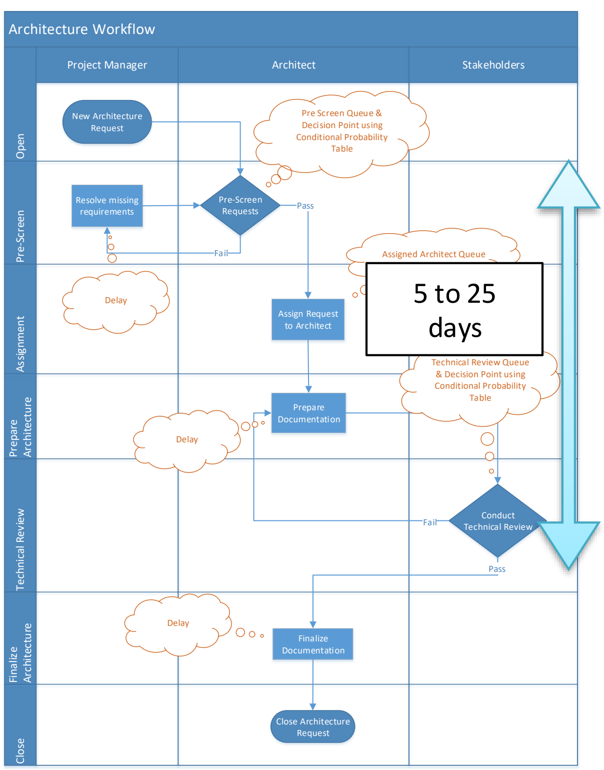


Figure 27. Evaluation Process

With a simulation model, you can quickly evaluate a few approaches to see if any of them are viable. You can then focus your attention on the ones that show the most promise.



We will consider these approaches:

- Hire more architects.
- Improve process efficiency, allowing existing architects to take on more than two projects at a time.
- Improve the probability of passing technical review by focusing on reasons why technical reviews fail.

In this section, we will to set up three experiments to evaluate each one of these approaches. We will show you how you can leverage simulation data in your decision-making process.

EXPERIMENT 1

Question: Can we shorten the time to pass technical review by adding architect resources while keeping the request volume unchanged from baseline levels?

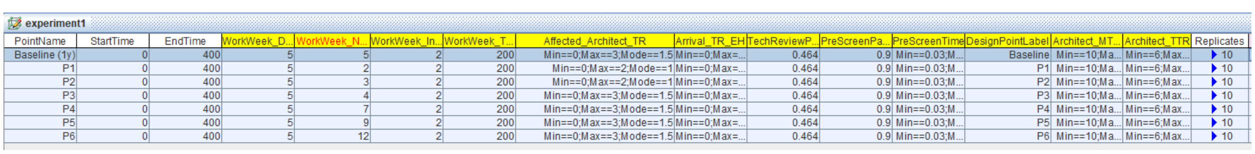
Anticipated Outcome: We will not see a reduction in the time it takes to pass technical review.

Inputs and Updates to Model:

The model does not require modification.

The experiment has six new design points, and factors are updated as shown in this table:

Factor	Baseline	P1	P2	P3	P4	P5	P6
StartTime	0						
EndTime	400						
WorkWeek_Duration	5						
WorkWeek_Num_Architect_TR	5	2	3	4	7	9	12
WorkWeek_IndividualTRCapacity	2						
WorkWeek_TROverloadThreshold	200						
Affected_Architect_TR	Min==0; Max==3; Mode==1.5	Min==0; Max==2; Mode==1	Min==0; Max==2; Mode==1				
Architect_MTBF	Min==10;Max==30; Mode==25						
Architect_TTR	Min==6;Max==10; Mode==7						
TechReviewPassProbability	0.464						
PreScreenPassProbability	0.9						
PreScreenTime	Min==0.03;Max==0.09; Mode==0.05						
Arrival_TR_EH	Min==0;Max==3; Mode==1.5						
DesignPointLabel	Baseline	P1	P2	P3	P4	P5	P6
Replicates	10						


experiment1

PointName	StartTime	EndTime	WorkWeek_D...	WorkWeek_N...	WorkWeek_In...	WorkWeek_T...	Affected_Architect_TR	Arrival_TR_EH	TechReviewP...	PreScreenPa...	PreScreenTime	DesignPointLabel	Architect_MT...	Architect_TTR	Replicates
Baseline (1y)	0	400	5	5	2	200	Min==0;Max==3;Mode==1.5	Min==0;Max=...	0.464	0.9	Min==0.03;M...	Baseline	Min==10;Ma...	Min==6;Max...	10
P1	0	400	5	2	2	200	Min==0;Max==2;Mode==1	Min==0;Max=...	0.464	0.9	Min==0.03;M...	P1	Min==10;Ma...	Min==6;Max...	10
P2	0	400	5	3	2	200	Min==0;Max==2;Mode==1	Min==0;Max=...	0.464	0.9	Min==0.03;M...	P2	Min==10;Ma...	Min==6;Max...	10
P3	0	400	5	4	2	200	Min==0;Max==3;Mode==1.5	Min==0;Max=...	0.464	0.9	Min==0.03;M...	P3	Min==10;Ma...	Min==6;Max...	10
P4	0	400	5	7	2	200	Min==0;Max==3;Mode==1.5	Min==0;Max=...	0.464	0.9	Min==0.03;M...	P4	Min==10;Ma...	Min==6;Max...	10
P5	0	400	5	9	2	200	Min==0;Max==3;Mode==1.5	Min==0;Max=...	0.464	0.9	Min==0.03;M...	P5	Min==10;Ma...	Min==6;Max...	10
P6	0	400	5	12	2	200	Min==0;Max==3;Mode==1.5	Min==0;Max=...	0.464	0.9	Min==0.03;M...	P6	Min==10;Ma...	Min==6;Max...	10

Figure 28. Screenshot of Experiment Window for Experiment 1

Conduct Experiment: We start the experiment by clicking Play.

SAS Simulation Studio ran for a few minutes, and the results were stored in the `\TechReview\results` directory. After the experiment was completed, the results were moved to the `\Experiment1` folder.

Analyze Results:

For the purpose of conducting this experiment, the contents of the files `\Experiment1\Close\Bucket_TR\result_tr_[0-6]_[0-9].sas7bdat` interest us the most.

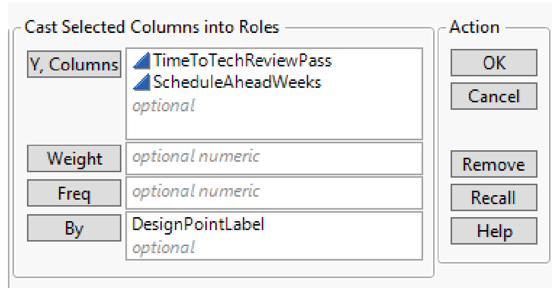


Figure 29. Selected Experiment 1 Columns

We prepare results for analysis by performing these steps:

1. Merge all files into a single JMP table.
2. Exclude rows completed after a one-year cutoff.
3. Create a distribution view of the TimeToTechReviewPass and ScheduleAheadWeeks columns, grouped by the DesignPointLabel column.
4. Review distribution in JMP.
   (Because Figure 31 is difficult to read, I've summarized relevant figures in the table below).

Metric	Baseline	P1	P2	P3	P4	P5	P6
Requests per year	168	141	167	167	167	167	167
Days to Tech Review Passing	6 to 26	12 to 55	7 to 26	6 to 26	6 to 26	6 to 26	6 to 26
Schedule Ahead Weeks	0	0 to 6	0	0	0	0	0

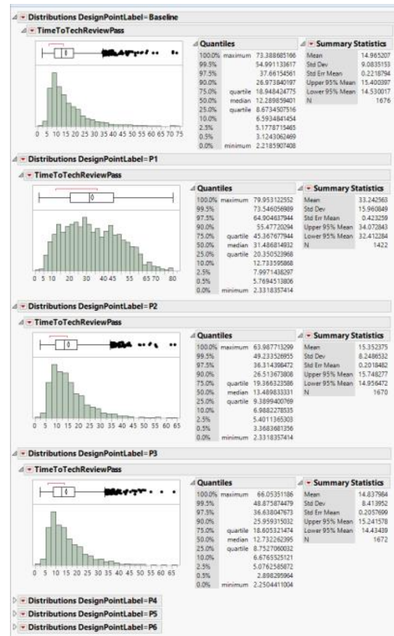


Figure 30. Experiment 1 Distribution

Review Actual Outcome

P4, P5, P6 - We did not see a <u>decrease</u> in the time to pass technical review when the number of architects was increased from 5 to 12.

P2, P3 - We did not see an <u>increase</u> in the time to pass technical review when the number of architects was reduced from five to three.

P1 - We saw an <u>increase</u> in the time to pass technical review when the number of architects was reduced from five to two. We also saw an increase in the request backlog of up to six weeks. Due to lack of available resources, we also saw a decrease in the number of completed requests per year from 168 to 141.

Conclusion

Our anticipated outcome was confirmed. If our current technical review volume remains the same, we will not shorten the time required to pass technical review by adding more architects. Furthermore, if the request volume remains the same, we can reduce the number of architects from five to three without impacting our turnaround time.

EXPERIMENT 2

Question: Can we shorten the time to pass technical review by increasing the workload capacity of each architect while keeping the request volume unchanged from baseline levels?

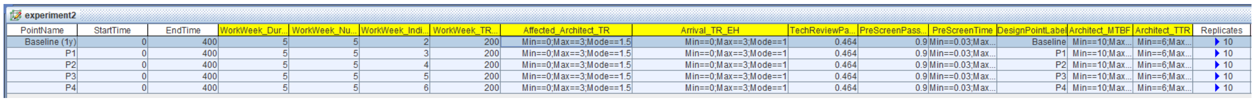
Anticipated Outcome: We will not see a reduction in the time it takes to pass technical review.

Inputs and Updates to Model:

The model does not require modification.

The experiment has four new design points and factors are updated as shown in this table:

Factor	Baseline	P1	P2	P3	P4
StartTime	0				
EndTime	400				
WorkWeek_Duration	5				
WorkWeek_Num_Architect_TR	5				
WorkWeek_IndividualTRCapacity	2	3	4	5	6
WorkWeek_TROverloadThreshold	200				
Affected_Architect_TR	Min==0; Max==3; Mode==1.5				
Architect_MTBF	Min==10;Max==30; Mode==25				
Architect_TTR	Min==6;Max==10; Mode==7				
TechReviewPassProbability	0.464				
PreScreenPassProbability	0.9				
PreScreenTime	Min==0.03;Max==0.09; Mode==0.05				
Arrival_TR_EH	Min==0;Max==3; Mode==1.5				
DesignPointLabel	Baseline	P1	P2	P3	P4
Replicates	10				



experiment2

PointName	StartTime	EndTime	WorkWeek_Dur...	WorkWeek_Nu...	WorkWeek_Indi...	WorkWeek_TR...	Affected_Architect_TR	Arrival_TR_EH	TechReviewPa...	PreScreenPass...	PreScreenTime	DesignPointLabel	Architect_MTBF	Architect_TTR	Replicates
Baseline (1y)	0	400	5	5	2	200	Min==0;Max==3;Mode==1.5	Min==0;Max==3;Mode==1	0.464	0.9	Min==0.03;Max...	Baseline	Min==10;Max...	Min==6;Max...	10
P1	0	400	5	5	3	200	Min==0;Max==3;Mode==1.5	Min==0;Max==3;Mode==1	0.464	0.9	Min==0.03;Max...	P1	Min==10;Max...	Min==6;Max...	10
P2	0	400	5	5	4	200	Min==0;Max==3;Mode==1.5	Min==0;Max==3;Mode==1	0.464	0.9	Min==0.03;Max...	P2	Min==10;Max...	Min==6;Max...	10
P3	0	400	5	5	5	200	Min==0;Max==3;Mode==1.5	Min==0;Max==3;Mode==1	0.464	0.9	Min==0.03;Max...	P3	Min==10;Max...	Min==6;Max...	10
P4	0	400	5	5	6	200	Min==0;Max==3;Mode==1.5	Min==0;Max==3;Mode==1	0.464	0.9	Min==0.03;Max...	P4	Min==10;Max...	Min==6;Max...	10

Figure 31. Screenshot of Experiment Window for Experiment 2

Conduct Experiment: We start the experiment by clicking Play.

SAS Simulation Studio ran for a few minutes and the results were stored in the `\TechReview\results` directory. After the experiment was completed, results were moved to the `\Experiment2` folder.

Analyze Results:

For the purpose of conducting this experiment, the contents of the files \Experiment1\Close\Bucket_TR\result_tr_[0-4]_[0-9].sas7bdat interest us the most.

We prepare results for analysis by performing these tasks:

1. Merge all files into a single JMP table.
2. Exclude rows completed after a one-year cutoff.
3. Create a distribution view of the TimeToTechReviewPass column, grouped by the DesignPointLabel column.
4. Review distribution in JMP.
   (Because Figure 35 is difficult to read, I've summarized relevant figures in the table below).

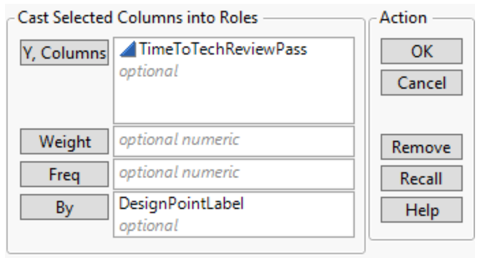


Figure 32. Selected Experiment 2 Columns

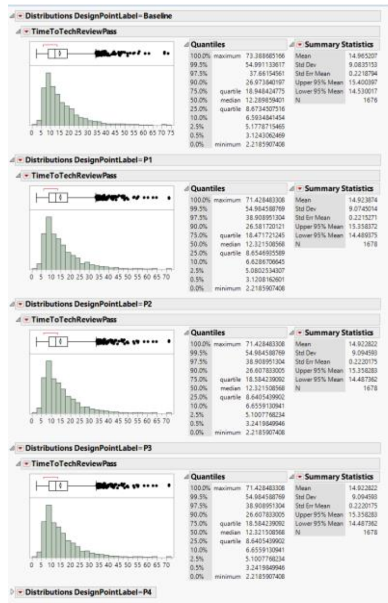


Figure 33. Experiment 2 Distribution

Metric	Baseline	P1	P2	P3	P4
Requests per year	168	168	168	168	168
Days to Tech Review Passing	6 to 26	6 to 26	6 to 26	6 to 26	6 to 26
Schedule Ahead Weeks	0	0	0	0	0

Review Actual Outcome:

P1-P4 - There was no change to the time it takes to pass technical review if an individual architect's workload capacity is increased.

Conclusion:

Our anticipated outcome was confirmed. With our current technical review volume, increasing the workload capacity of individual architects will not make any impact on time it takes to pass technical review.

EXPERIMENT 3

Question: Can we shorten the time to pass technical review by improving the technical review pass probability, while keeping request volume unchanged from baseline levels?

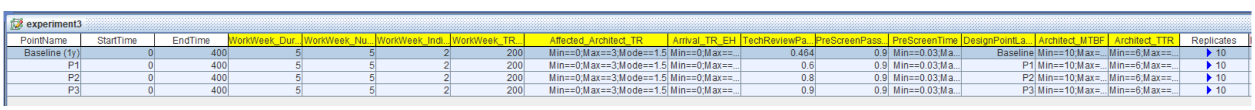
Anticipated Outcome: We will see a shorter time to pass technical review, because fewer attempts at technical review will be made.

Inputs and Updates to Model:

The model does <u>not</u> require modification.

The experiment has three new design points, and factors are updated as shown in this table:

Factor	Baseline	P1	P2	P3
StartTime	0			
EndTime	400			
WorkWeek_Duration	5			
WorkWeek_Num_Architect_TR	5			
WorkWeek_IndividualTRCapacity	2			
WorkWeek_TROverloadThreshold	200			
Affected_Architect_TR	Min==0; Max==3; Mode==1.5			
Architect_MTBF	Min==10;Max==30; Mode==25			
Architect_TTR	Min==6;Max==10; Mode==7			
TechReviewPassProbability	0.464	.6	.8	.9
PreScreenPassProbability	0.9			
PreScreenTime	Min==0.03;Max==0.09; Mode==0.05			
Arrival_TR_EH	Min==0;Max==3; Mode==1.5			
DesignPointLabel	Baseline	P1	P2	P3
Replicates	10			



experiment3

PointName	StartTime	EndTime	WorkWeek_Dur...	WorkWeek_Nu...	WorkWeek_Indi...	WorkWeek_TR...	Affected_Architect_TR	Arrival_TR_EH	TechReviewPa...	PreScreenPass...	PreScreenTime	DesignPointLa...	Architect_MTBF	Architect_TTR	Replicates
Baseline (1y)	0	400	5	5	2	200	Min==0;Max==3;Mode==1.5	Min==0;Max==...	0.464	0.9	Min==0.03;Ma...	Baseline	Min==10;Max=...	Min==6;Max==...	10
P1	0	400	5	5	2	200	Min==0;Max==3;Mode==1.5	Min==0;Max==...	0.6	0.9	Min==0.03;Ma...	P1	Min==10;Max=...	Min==6;Max==...	10
P2	0	400	5	5	2	200	Min==0;Max==3;Mode==1.5	Min==0;Max==...	0.8	0.9	Min==0.03;Ma...	P2	Min==10;Max=...	Min==6;Max==...	10
P3	0	400	5	5	2	200	Min==0;Max==3;Mode==1.5	Min==0;Max==...	0.9	0.9	Min==0.03;Ma...	P3	Min==10;Max=...	Min==6;Max==...	10

Figure 34. Screenshot of Experiment Window for Experiment 3

Conduct Experiment: We start the experiment by clicking Play.

SAS Simulation Studio ran for a few minutes and the results were stored in the `\TechReview\results` directory. After the experiment was completed, the results were moved to the `\Experiment3` folder.

Analyze Results:

For the purpose of conducting this experiment, the contents of the files `\Experiment1\Close\Bucket_TR\result_tr_[0-3]_[0-9].sas7bdat` interest us the most.

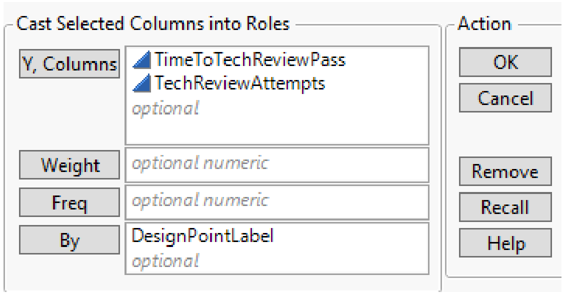


Figure 35. Selected Experiment 3 Columns

We prepare results for analysis by performing these tasks:

1. Merge all files into a single JMP table.
2. Exclude rows completed after a one-year cutoff.
3. Create a distribution view of the TimeToTechReviewPass and TechReviewAttempts columns, grouped by the DesignPointLabel column.
4. Review distribution in JMP.
   (Because Figure 39 is difficult to read, I've summarized relevant figures in the table below).

Metric	Baseline	P1	P2	P3
Requests per year	168	170	171	172
Days to Tech Review Passing	6 to 26	6 to 21	6 to 15	5 to 14
Tech Review Attempts	1 to 4	1 to 3	1 to 2	1

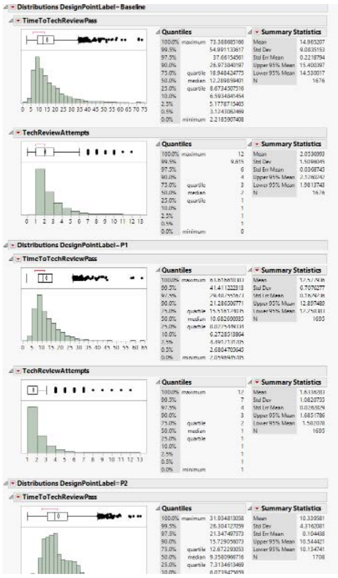


Figure 36. Experiment 3 Distribution

Review Actual Outcome:

P1-P3 – As the probability of passing the technical review increased, the number of attempts and the time it took to pass technical review decreased.

Conclusion:

Our anticipated outcome was confirmed. Increasing the probability of passing each technical review will decrease the technical review pass turnaround time.

EXPERIMENT SUMMARY

In the three experiments we just ran, we were able to analyze over 30,000 records and decades worth of data in a span of a few minutes. We were able to identify a viable strategy and rule out ideas that are not worth pursuing. More importantly, our decisions are backed up by data and the expected results are known. This allows us to look at business problems with a cost/benefit mindset instead of with a gut feeling.

TIPS

Getting started is usually the hardest part. Here are a few things that helped us break ground when we used SAS Simulation Studio to model our business process.

- Identify and gather required information before building your model.
- Workflows and process maps are great tools for identifying key areas within your process.
- Planning allows you to build models faster and with higher accuracy because required inputs and outputs are easier to identify and track.

As you comb through your process and workflows, the data necessary to make your model work will become apparent to you. If you are working with an existing process, some of the data might already be collected; otherwise you'll need to make educated guesses.

Remember, <u>output quality depends on input quality.</u>

Resist the temptation to make your model and workflow too detailed. Instead, focus on your use case and adjust the detail accordingly. Try consolidating small steps if the result will still be the same. Consolidation will make your model easier to work with and will reduce overall complexity.

For example, if our goal is simulating the amount of time a customer spends in a check-out line, it might be useful to know how long they have been shopping, but it might not be necessary to track their movements within the store prior to entering check-out queue.

Your first goal is to build a simulation model that you can use as a baseline for your future experiments. Remember to validate the model against real-world data before trusting simulated outputs. During model validation, the simulation output should produce data that comes "*close enough*" to real-world results. The "*close enough*" value will vary depending on your application.

After you baseline model is confirmed to be accurate, experimentation can begin. Don't be alarmed if you are forced to introduce new model features or variables during the experimentation phase. This is an iterative design approach, and with time you'll become more proficient in it.

TRAINING

If you are not familiar with building models in SAS Simulation Studio, I highly recommend taking the training course from SAS called Discrete-Event Simulation with SAS Simulation Studio. This course is for analysts who need to use discrete-event simulation in order to model complex systems that are difficult or impossible to model using traditional analytical techniques. Discrete-event simulation models dynamic systems whose state changes only when distinct, discrete events occur. The simulation models can then be used to look at various changes to the processes to determine the impacts those changes can have.

During the course, you will learn how to build, run, and analyze discrete-event simulation models using SAS Simulation Studio software. Specifically, you will learn how to perform these tasks:

- Use the discrete-event simulation study steps to create a high-level simulation model.
- Use discrete-event simulation modeling concepts to develop models of dynamic systems.
- Model complexity through the use of conditional logic, schedules, resource constraints, and more.
- Verify and validate the accuracy and appropriateness of simulation models.

CONCLUSION

As you have seen for yourself, after you have a good model with which to work, running experiments in SAS Simulation Studio and analyzing results in JMP is quick and intuitive, and provides real business value.

By backing up your assumptions with data, you will be able to craft more persuasive business strategies, know your performance targets, and know what to expect in return. You can plan for changes or identify breaking points in your process in advance.

Knowledge is power, and with SAS tools at your disposal, you have the Power to Know

APPENDIX A

Here are some up-close images of the simulation model in Figure 3.

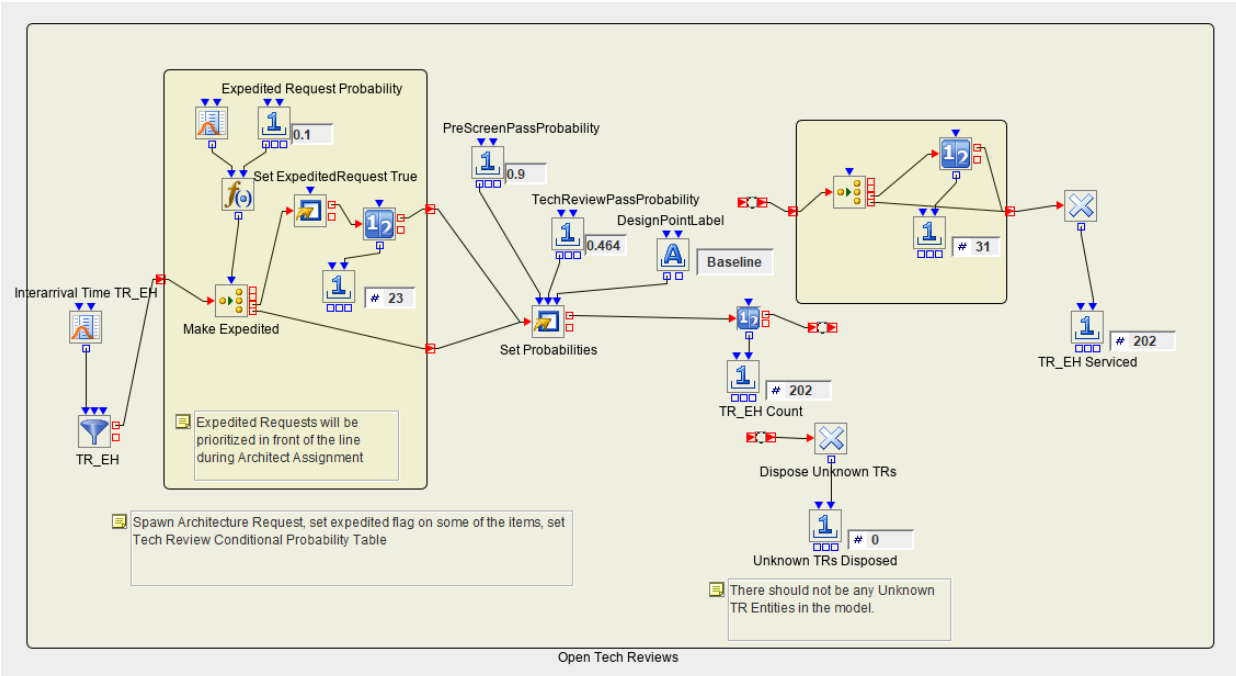


Figure 37. Block 1: New Architecture Requests Block

Here is a brief description of each block and the role it plays in the model:

Block 1 - New Architecture Requests Block – Tracks creation, disposal, and priority of new architecture requests.

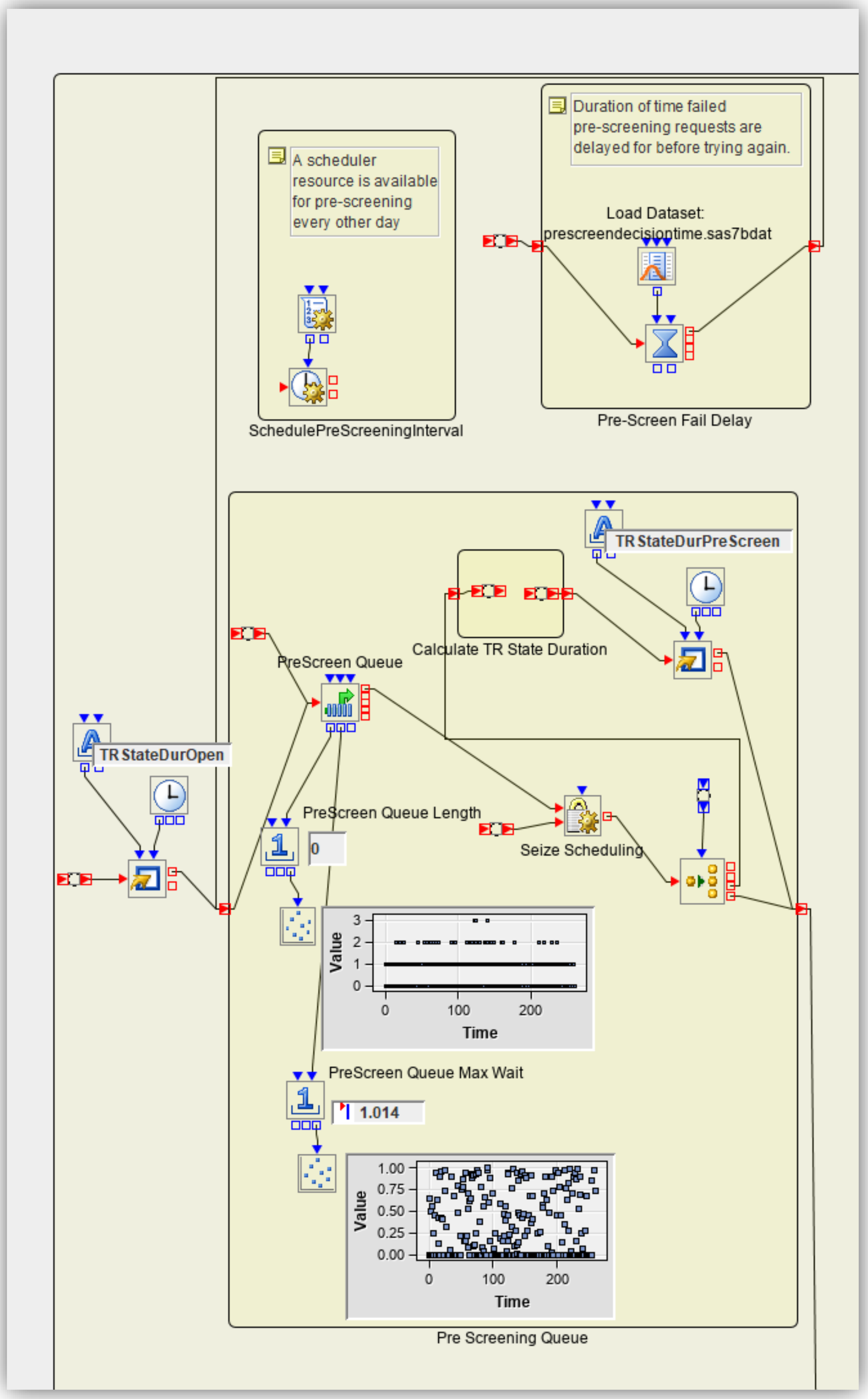


Figure 38. Block 2: Pre-Screening Queue and Delay Block

Block 2 - Pre-Screening Queue and Delay Block – Tracks the pre-screening queue, seizure of the scheduling resource, and simulation of the delay when the architecture request fails pre-screening.

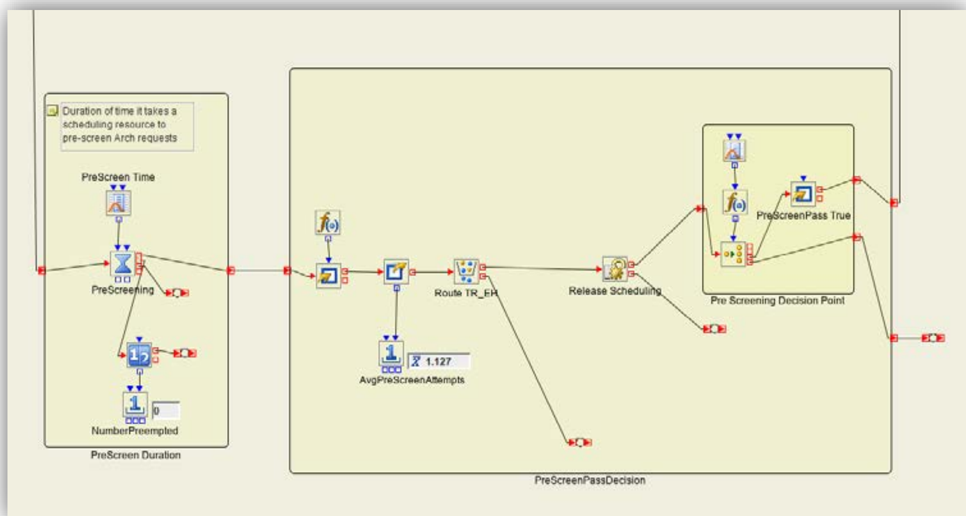


Figure 39. Block 3: Pre-Screen Decision Point

Block 3 - Pre-Screen Decision Point – Loads conditional probability data set, tracks pre-screening attempts and decisions, and releases a scheduling resource after recording a decision.

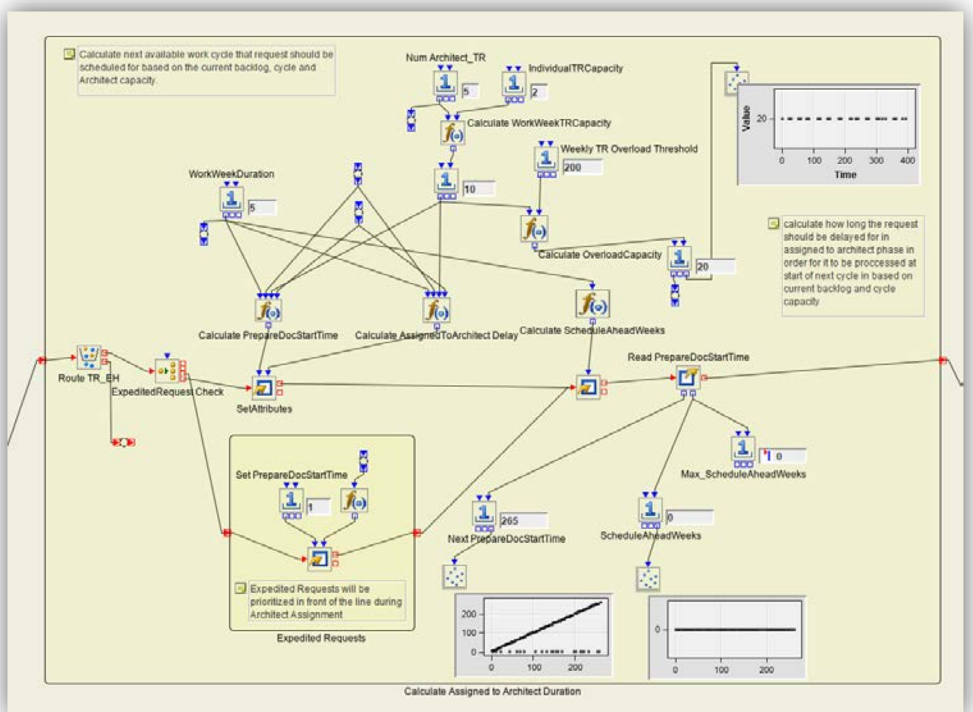


Figure 40. Block 4: Calculate Assigned to Architect Delay

Block 4 - Calculate Assigned to Architect Delay – Calculates how many weeks ahead a request will be scheduled, based on request priority, current backlog, and process capacity.

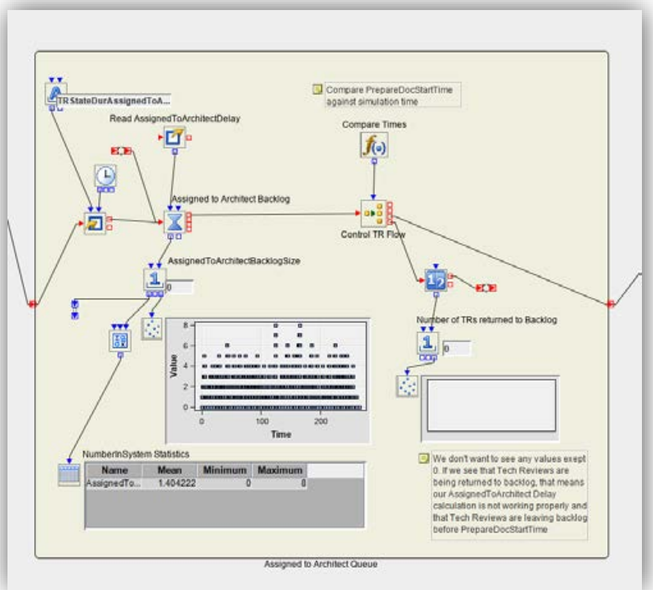


Figure 41. Block 5: Assigned to Architect Queue

Block 5 - Assigned to Architect Queue – Holding area for Architecture requests, also known as the Architecture Request Backlog.

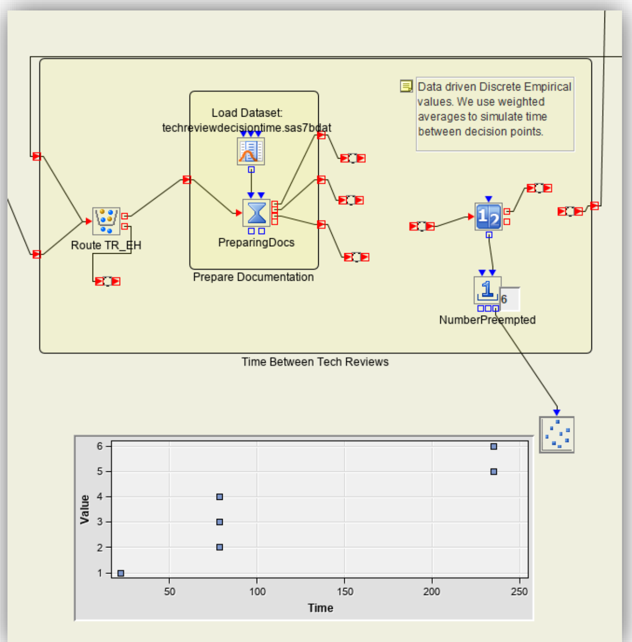


Figure 42. Block 6: Preparing Documentation

Block 6 - Preparing Documentation – Using an external data set, the delay duration is calculated. This delay represents how long an architect will work on preparing documentation before attempting to bring it to technical review.

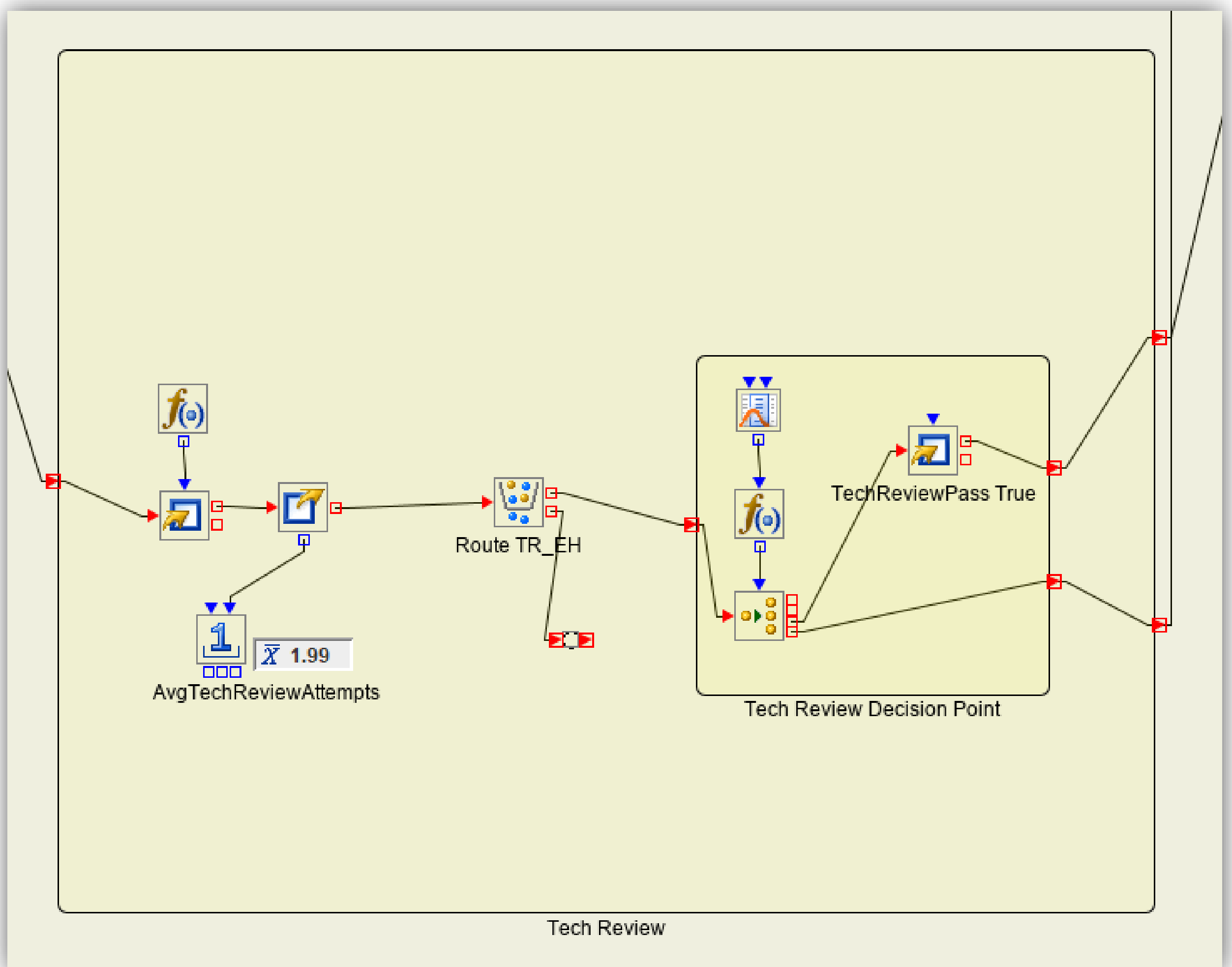


Figure 43. Block 7: Tech Review Decision Point

Block 7 - Tech Review Decision Point - Loads conditional probability data set, tracks technical review attempts and decisions.

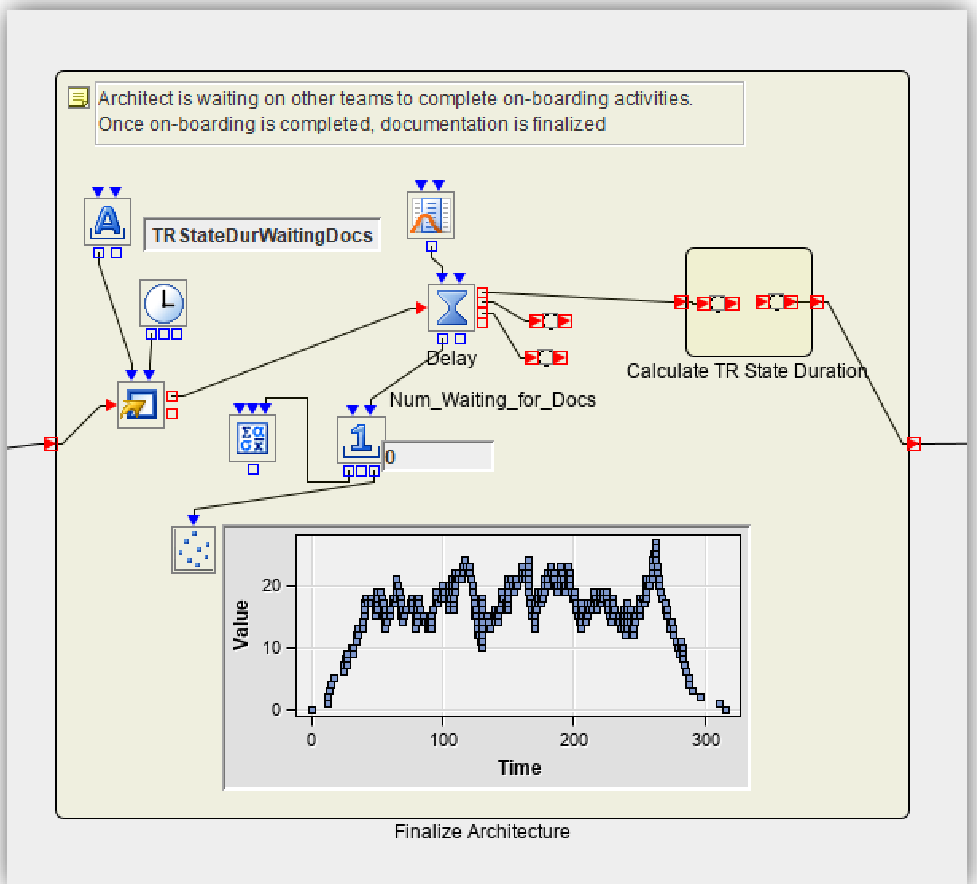


Figure 44. Block 8: Finalizing Documentation

Block 8 – Finalizing Documentation - Simulates the delay experienced by the architect while on-boarding activities are completed by other teams.

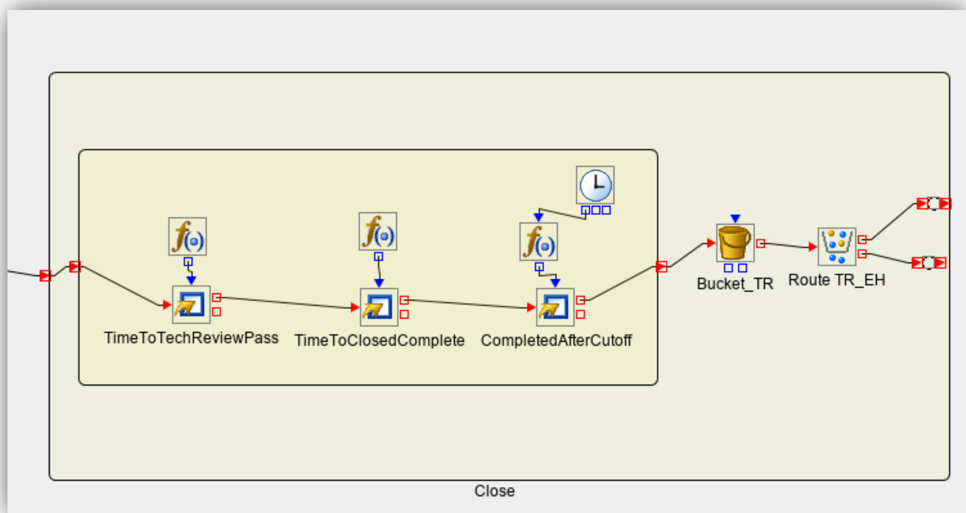


Figure 45. Bock 9: Closing Block

Block 9 – Closing Block – Additional metrics are calculated, and request is sent to the disposer.

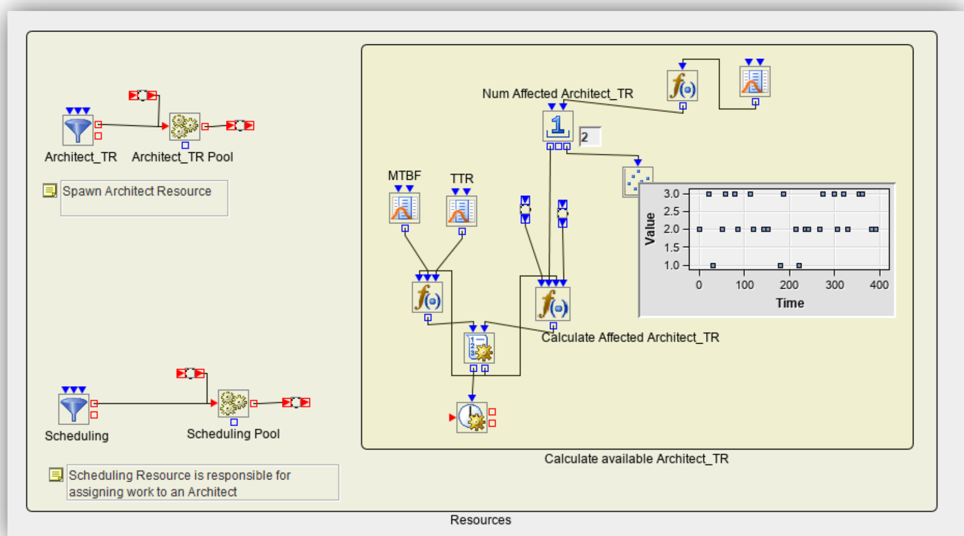


Figure 46. Block 10: Resourcing Block

Block 10 - Resourcing Block – Supports pre-emption or re-assignment of work when the architect becomes unavailable. Supports seizing and releasing of available architects based on experiment factors.

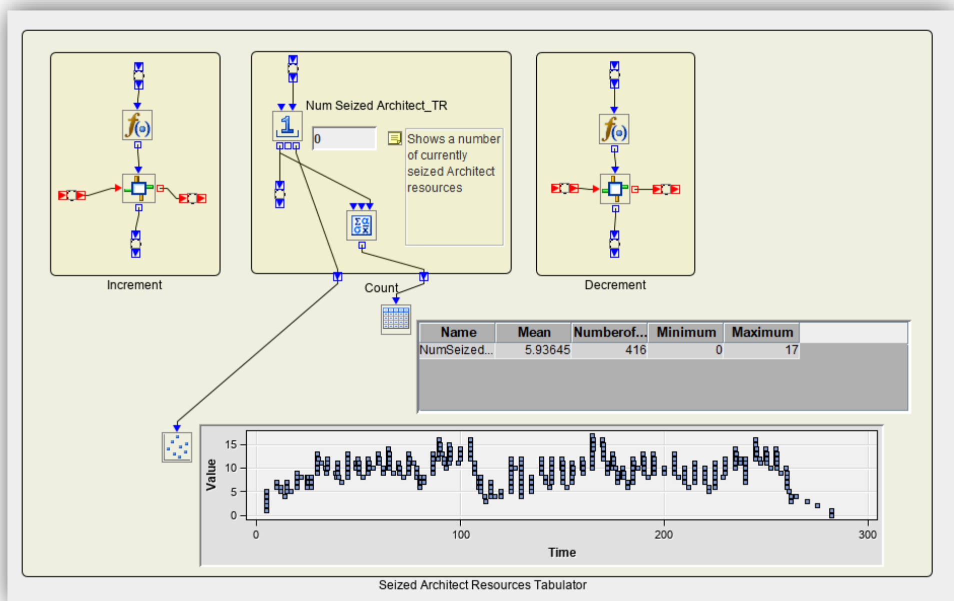


Figure 47. Block 11: Seized Architect Tabulator

Block 11 – Seized Architect Tabulator – Tracks the number of currently seized architect resources.

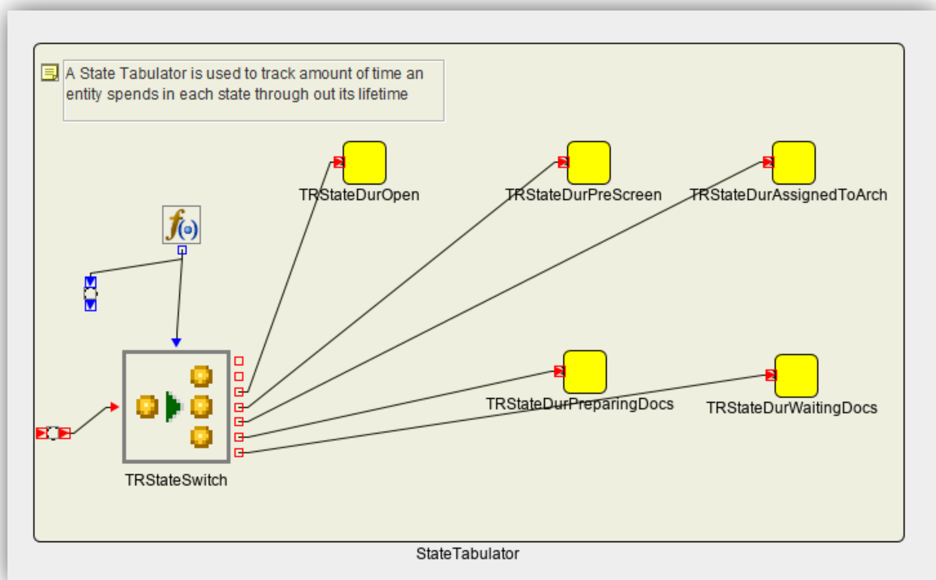


Figure 48. Block 12: Arch Request Stat Tabulator

Block 12 – Arch Request Stat Tabulator – Calculates durations of architecture request states for every entity.

ACKNOWLEDGMENTS

I want to acknowledge Rida Ahmed, Dan Obermiller, Elliot Inman, and Dan Kelly for their contributions to this work.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Dave Manet
SAS Institute Inc.
Dave.Manet@sas.com
www.SAS.com