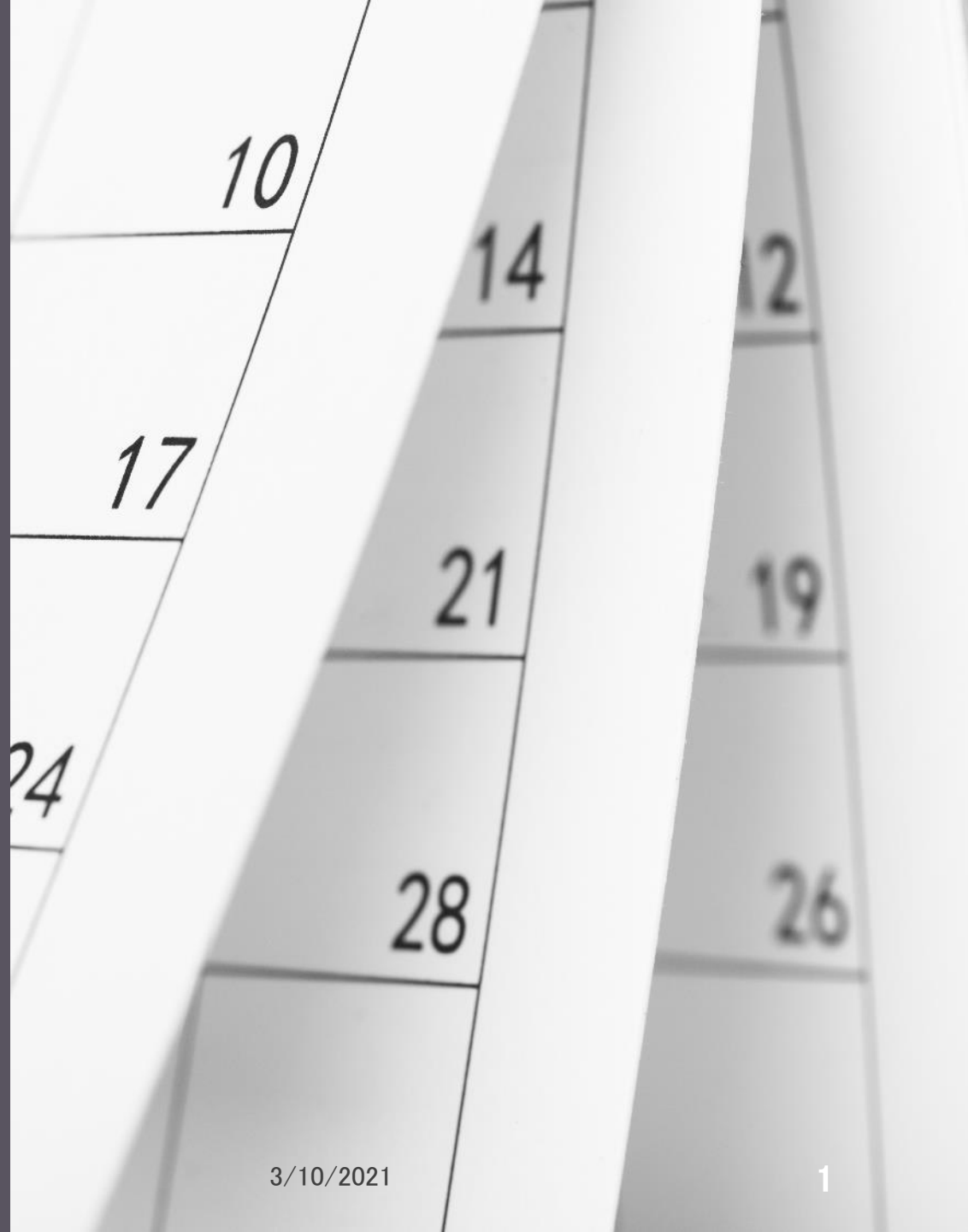
PROJECT CRASHING USING LINEAR PROGRAMMING

Esma Gunes

Ben Prescott

Robin Singh



Agenda

Introductions & Project Background

Challenges & Assumptions

Approach & Model Formulation

Results, Limitations and Future

INTRODUCTIONS & PROJECT BACKGROUND

Esma Gunes

Ben Prescott

Robin Singh

The Team

Project Background

This idea originated from project delivery experience within the technology consulting space, specifically with cloud projects. It is common for projects to begin under the assumption that the duration of the project is dictated by the task/milestone efforts.

However, this is commonly not the case as many clients may impose a soft or hard deadline for a variety of reasons, which requires a shift in how the project would be delivered. This leads to a “fire drill” in finding ways to support the project.

Business Challenge

Consulting Services organizations face issues with effectively balancing customer expectations regarding sales cycles and project delivery timelines. Once a project is sold and staffed, each project follows a (mostly) standard process of defining a project backlog, associated hours estimates/duration with each task, and estimated duration based on the tasks.

Inefficient project deadlines imposed after the project has begun typically leads to:

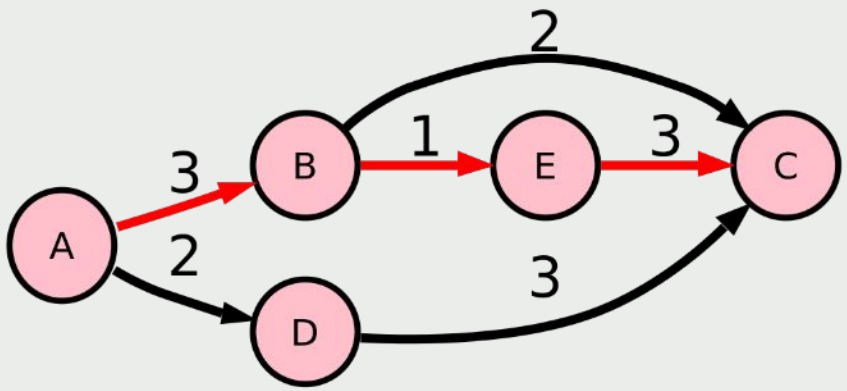
- The project delivery structure needing to shift to meet the newly determined deadline
- Determining creative ways to reduce overall project duration to meet the deadline
- Assigning additional resources to support the effort, thus increasing internal cost/reduced deal margin
- The need to keep additional costs to a minimum as to still make a profit

Motivation & Objective

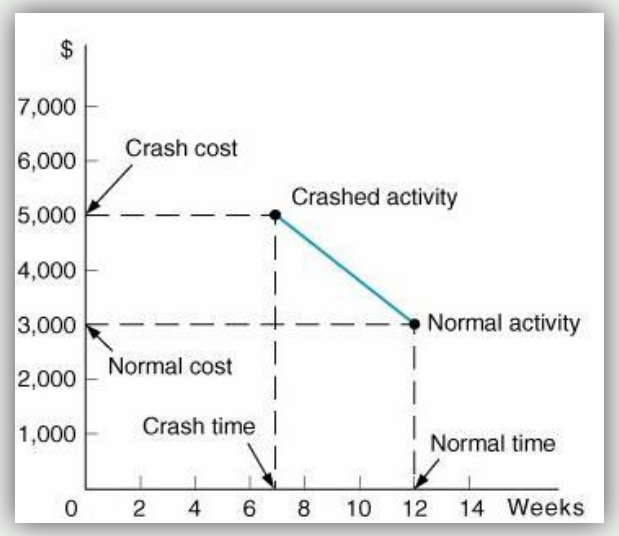
Our project was motivated by real-world issues with the need to condense timelines, move faster through a project, and scale out a project team to meet client requirements. Our experience has been that when a project timeline needs to shift up, many options come to the table that might lead to crashing the project.

Part of determining a solution requires the project team working in tandem with Resource Management to determine who on the team has availability, the proper skills, identify tasks that can fast tracked, etc., which is generally a slow, manual process.

The main objective of the project is to determine which tasks within a project plan can be crashed, by how many days they can be crashed, and the total cost added to the project to crash the respective tasks while hitting a specific deadline.



Understanding Project Crashing



The purpose behind *project crashing* or *crashing a project* is to shorten the overall project duration to hit a certain deadline by reducing the duration of each individual task.

This becomes an exercise of determining which tasks to crash, by how long, and how much the added internal cost is of adding resources to help with those tasks.

The crashing effort typically starts with creating a Program Evaluation Review Technique (PERT) diagram/chart, then determining the project's *Critical Path.*

The critical path determines the longest stretch of activities that determining the minimum time required to complete the project. Certain projects that are crashed may shift the initial critical path.

CHALLENGES & ASSUMPTIONS

Challenges

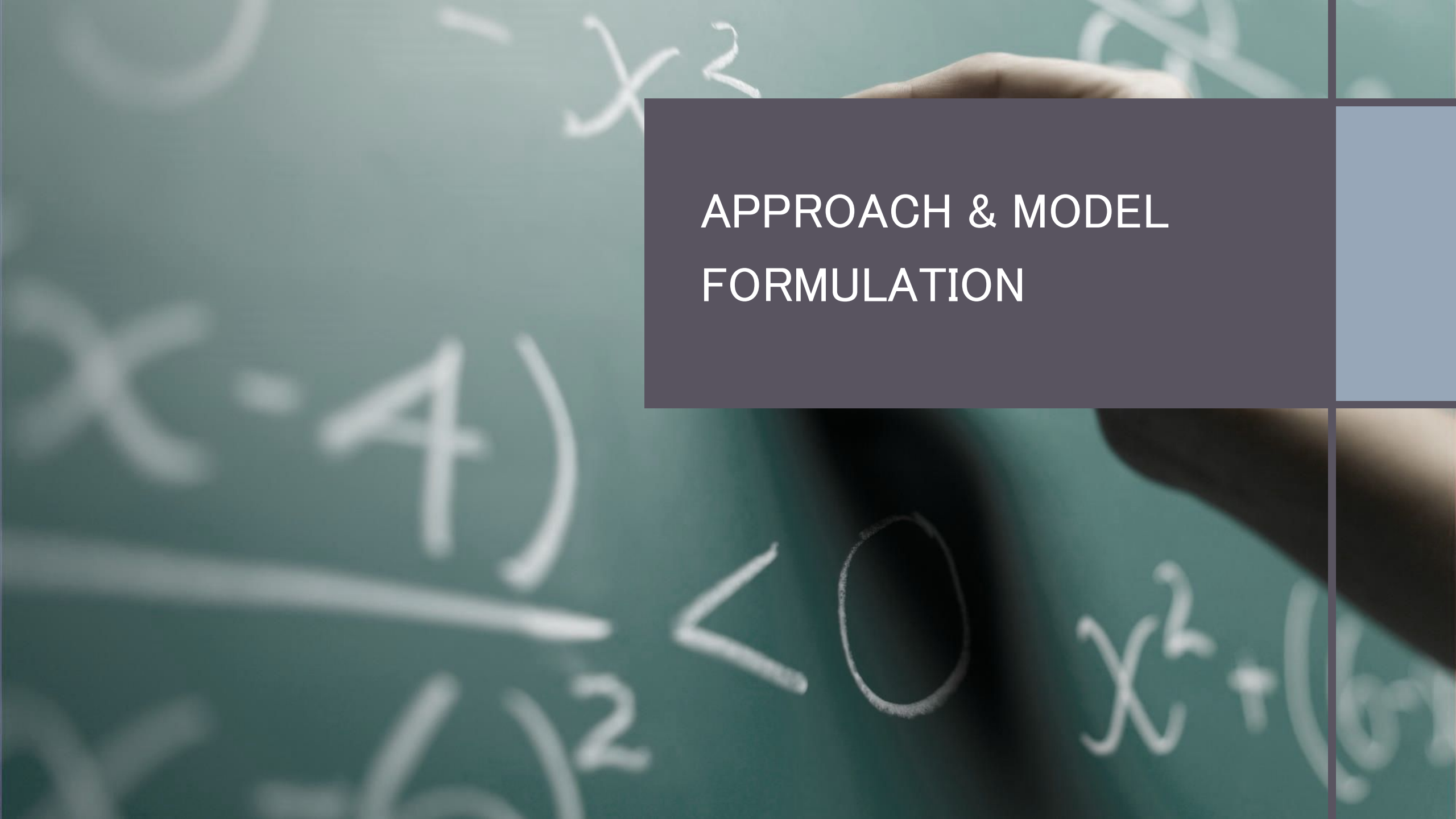
Some challenges encountered during this project were:

- **Understanding Project Management concepts**
    - Determining how projects are typically ran, common problems encountered, what Critical Path and PERT charts are, etc.
- **Gathering required project information**
    - Obtaining historical data around individual task durations and examples of crashed projects required some improvising
- **Creating the additional project requirements**
    - Which tasks can be crashed, how much they can be crashed, updating the information after removal of some sensitive items, etc.
- **Formulating the exact LP model**
    - The model includes many decision variables – 25 to be exact – and requires some mildly confusing constraints
    - Which tool to use to formulate the model to make updates as easy as possible

Assumptions

We've also included some assumptions with our project, including:

- **Project Resources**
  - Resources are available full time (40 hours / week)
  - Resources bill a consistent 8-hour day regardless
  - Delivery resources fall into one of two roles: Senior Consultant or Consultant
  - Senior Consultants and Consultants have standard hourly bill rates
- **Project Selection**
  - The project selected is one that has repeatable/consistent task efforts
  - The crash potentials for each task are known or have been determined prior to modeling
  - The project follows a standard project plan and is not ran in an ad-hoc manner



APPROACH & MODEL FORMULATION

Approach Overview

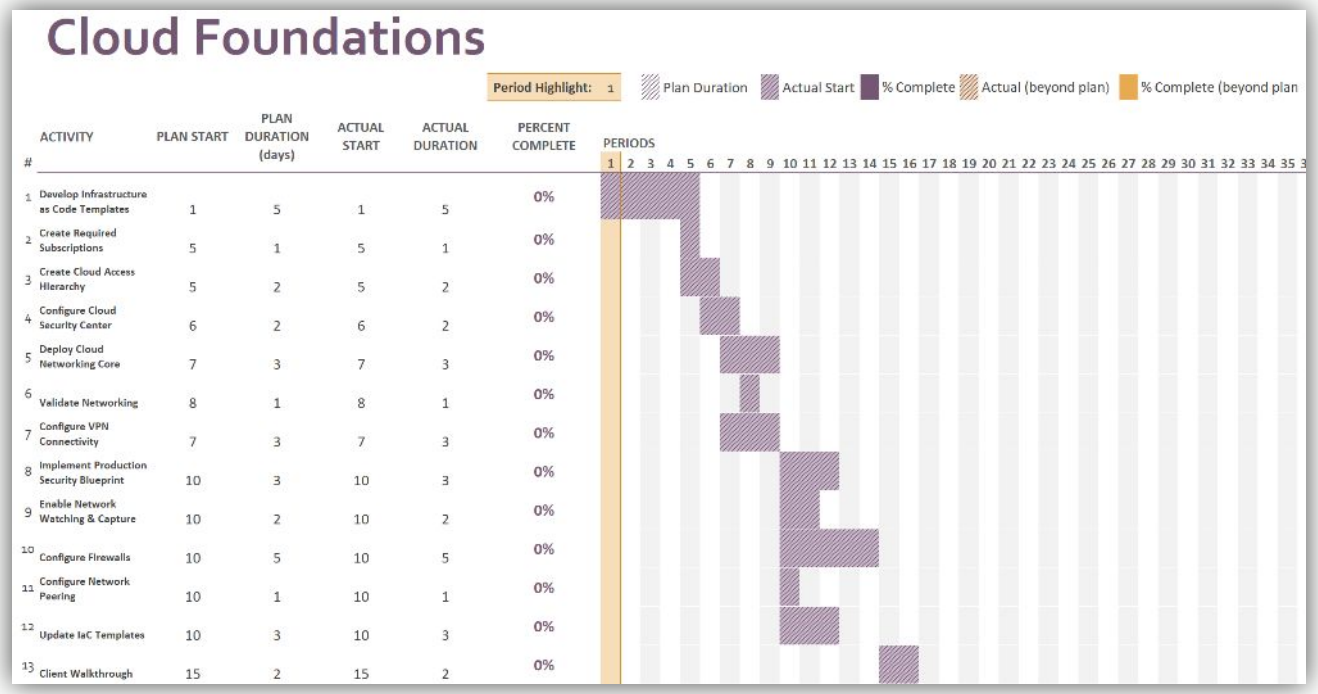
We started this project by using a real-world example of a project from Ben's company, which focuses on a *cloud foundation*. Cloud foundation projects are usually turnkey solutions, requiring little to no custom modification. Certain information has been removed/changed to keep from displaying sensitive company information, but the general tasks and project timelines remain intact.

Before we could create our LP formulation, we had to further define project specifics. We followed the below process:

1. Select a project
2. Create project plan to determine tasks/project duration
3. Define individual task crash potentials in days
4. Calculate each task's crash cost per day
5. Create PERT chart to visualize project network, task dependencies and critical path
6. Create LP model

Project Planning

Following our approach, we started by selecting the Cloud Foundations project to model against, then creating a project plan. Our goal of this model will be to reduce the project duration from 16 days down to 11 days, removing one business week's worth of time.



#	ACTIVITY	PLAN START	PLAN DURATION (days)	ACTUAL START	ACTUAL DURATION	PERCENT COMPLETE
1	Develop Infrastructure as Code Templates	1	5	1	5	0%
2	Create Required Subscriptions	5	1	5	1	0%
3	Create Cloud Access Hierarchy	5	2	5	2	0%
4	Configure Cloud Security Center	6	2	6	2	0%
5	Deploy Cloud Networking Core	7	3	7	3	0%
6	Validate Networking	8	1	8	1	0%
7	Configure VPN Connectivity	7	3	7	3	0%
8	Implement Production Security Blueprint	10	3	10	3	0%
9	Enable Network Watching & Capture	10	2	10	2	0%
10	Configure Firewalls	10	5	10	5	0%
11	Configure Network Peering	10	1	10	1	0%
12	Update IaC Templates	10	3	10	3	0%
13	Client Walkthrough	15	2	15	2	0%

Crash Determination

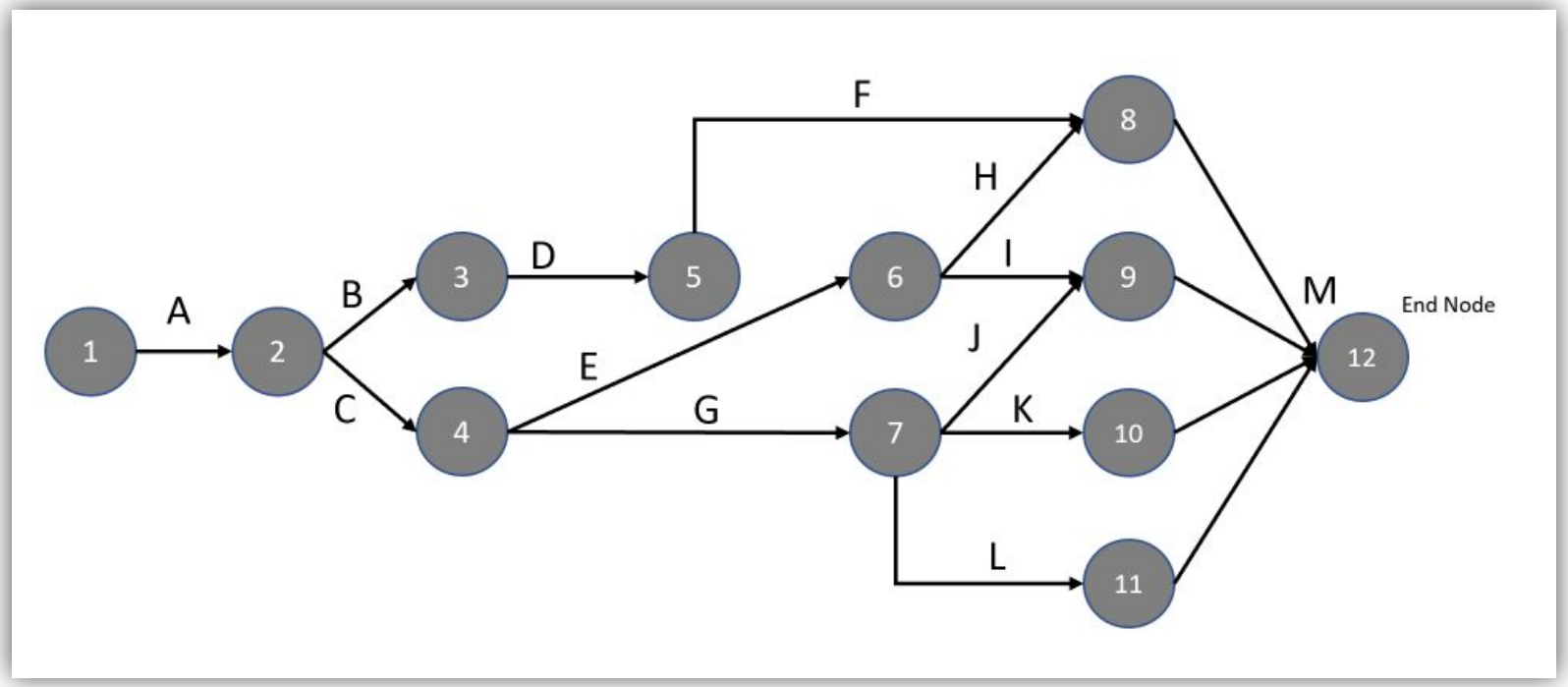
Once the project plan was defined, we built a spreadsheet to model our activities, dependencies, the standard duration of tasks and the crash durations. From here we were able to determine the crash potentials for each task and the crash cost per day, which will be used in our model.

Name	Activities	Predecessor	Normal Time (days)	Crash Time	Normal Cost	Crash Cost	Activity Time Difference (Normal - Crash)	Added Cost for	Crash cost/time
Develop Infrastructure as Code Templates	A		5	2	11000	17600	3	6600	2200
Create Required Subscriptions	B	A	1	0.5	2000	3000	0.5	1000	2000
Create Cloud Access Hierarchy	C	A	2	1	4000	6000	1	2000	2000
Configure Cloud Security Center	D	B	2	1	4000	6000	1	2000	2000
Deploy Cloud Networking Core	E	C	3	2	6600	8800	1	2200	2200
Validate Networking	F	D	1	0.8	2200	2640	0.2	440	2200
Configure VPN Connectivity	G	C	3	1.5	6000	9000	1.5	3000	2000
Implement Production Security Blueprint	H	E	3	2	6600	8800	1	2200	2200
Enable Network Watching & Capture	I	E	2	1	4000	6000	1	2000	2000
Configure Firewalls	J	G	5	2	11000	17600	3	6600	2200
Configure Network Peering	K	G	1	0.5	2200	3300	0.5	1100	2200
Update IaC Templates	L	G	3	1	6000	10000	2	4000	2000
Client Walkthrough	M	F,H,I,J,K,L	2	1	4400	6600	1	2200	2200

Role	Hourly Rate
Senior Consultant	275
Consultant	250

PERT Chart Creation

We then created a PERT chart to visualize the project's network model. This also helps in quickly finding dependencies between tasks and determining which tasks occur in which order.



Model Formulation

Objective Function:

$$MINIMIZE: \sum_{a=1}^{n} D_a C_a$$

Variables:

$C_a = specific\ task's\ crash\ cost\ per\ day$
$a = specific\ project\ task\ number$
$n = specific\ task's\ normal\ duration$

Decision Variables:

$D_a = number\ of\ days\ the\ specific\ task\ will\ be\ crashed$
$E_t = when\ an\ event\ will\ occur\ within\ the\ project$

Constraints:

$D_a \leq maximum\ crash\ potential\ for\ specific\ task$

$E_t \geq n - D_a + E_{t\_1}\}\ duration\ of\ each\ task\ in\ relation\ to\ previous\ dependent\ tasks's\ occurence$

$E_{12} \leq project\ deadline\ requirement\ (in\ days)$

$E_1 = 0$

RESULTS & LIMITATIONS

Model Results

The Goal was to reduce our overall project duration from 16 days to 11 days, and also minimizing the cost.

Objective (minimizing cost): $14,800 □ cost to add resources to hit the deadline

Tasks to be crashed:

- Task 1 (Develop Infrastructure as Code Templates): crashed by 2 days
- Task 3 (Create Cloud Access Hierarchy): crashed by 1 day
- Task 7 (Configure VPN Connectivity): crashed by 1 day
- Task 10 (Configure Firewalls): crashed by 1 day
- Task 12 (Update IaC Templates): crashed by 1 day
- Task 13 (Client Walkthrough): crashed by 1 day

Accomplishments

Name	Activities	Predecessor	Normal Time (days)	Crash Time	Normal Cost	Crash Cost	Activity Time Difference (Normal - Crash)	Added Cost for	Crash cost/time
Develop Infrastructure as Code Templates	A		5	2	11000	17600	3	6600	2200
Create Required Subscriptions	B	A	1	0.5	2000	3000	0.5	1000	2000
Create Cloud Access Hierarchy	C	A	2	1	4000	6000	1	2000	2000
Configure Cloud Security Center	D	B	2	1	4000	6000	1	2000	2000
Deploy Cloud Networking Core	E	C	3	2	6600	8800	1	2200	2200
Validate Networking	F	D	1	0.8	2200	2640	0.2	440	2200
Configure VPN Connectivity	G	C	3	1.5	6000	9000	1.5	3000	2000
Implement Production Security Blueprint	H	E	3	2	6600	8800	1	2200	2200
Enable Network Watching & Capture	I	E	2	1	4000	6000	1	2000	2000
Configure Firewalls	J	G	5	2	11000	17600	3	6600	2200
Configure Network Peering	K	G	1	0.5	2200	3300	0.5	1100	2200
Update IaC Templates	L	G	3	1	6000	10000	2	4000	2000
Client Walkthrough	M	F,H,I,J,K,L	2	1	4400	6600	1	2200	2200

Role	Hourly Rate
Senior Consultant	275
Consultant	250

Limitations

- The limitation of not being able to flex resources more or less than eight hours a day
- The decision-making takes place in the Project team.
- The Project Duration could be further reduced but we have to spent more money to achieve that task.

Future Work

Ideas for the future plan;

Pairing this model with another model called Resource Availability Cost Problem (RACP)

- The goal of the RACP is to minimize the total resource cost on a project
- Determining the resources to hit the deadline