**Agile Sprint Success Prediction Modeling**

By Kunal Nain

CS 558

May 16, 2019

**Table of Contents**

1. INTRODUCTION
   1. Background and Problem Description
   2. Simulation Considerations and Limitations
2. BASIC DATA ELICITATION
   1. Random Parameter Generation
   2. Model Parameters Boolean Specification
3. MODEL CONSTRUCTION, VERIFICATION, AND VALIDATION
4. MODEL DESIGN
5. RESULT ANALYSIS
6. CONCLUSION
7. APPENDIX
8. INTRODUCTION

This simulation study tries to predict the probability of success for an agile sprint. Agile methodology is used by almost all major companies to organize and conduct software development. To make the most out of a sprint, companies adhere to several standards and recommendations. Some of these recommendations include effective sprint length, story point estimation etc. The aim of this study is to present an argument for the most important parameters in sprint success and develop an equation. This can be beneficial to the corporate sector as it can provide a tool for the stakeholders to decide whether their plan is adhering to the industry standards and detect if there are any red flags.

1. BACKGROUND AND PROBLEM DESCRIPTION

Despite immense effort from the stakeholders, there is no industry standard tool to predict whether a sprint plan has been effectively designed. Every company’s culture is different and thus every sprint plan has to be unique. There are several factors like nature of job, expertise of the team etc. that separate teams and their sprints from each other. The equation helps the team to validate their sprint plan and prevents any catastrophic project planning disaster.

1. SIMULATION CONSIDERATIONS AND LIMITATIONS

The simulation considers three major factors which are team size, sprint length, and story point estimation. There are several sub factors that are considered by the study to decide upon the chosen factors. All of the three factors are listed below along with the subfactors:

1. Team Size
   1. Nature of work
   2. Size of company
   3. Software project
2. Sprint Length
   1. Pre and Post Sprint Planning
   2. Buffer time
   3. Team type (Hardware / Software)
3. Story Point Estimation
   1. Task
   2. Technology
   3. Estimation Accuracy

A close up of a logo

Description automatically generated

The study has its own limitation and data collection is the biggest one. Data collection from agile scrum boards is the biggest challenge as the required data for verification and validation is not publicly available. It contains high amounts of sensitive information regarding ongoing or past projects for the companies and this data is usually kept private by the company for internal use. Companies employing agile protect this information, so no open source data is available. Solutions to this problem is using scientific methods to elicit data. However, there is an online application called Jumble for Jira which generates sample data. This sample data can be used to verify and validate the model but there is no evidence that the sample data for verification and validation represents a data set from the industry.

1. BASIC DATA ELICITATION

Data elicitation will be done using random number generator. After the numbers have been generated the model parameters will need to be assigned their associated tag. The tag for the model variables in this context are Booleans. The model uses three primary parameters: team size, sprint length, and story point estimation. The range and the Boolean cutoff values for the parameters are discuss below.

1. RANDOM PARAMETER GENERATION

Data is randomly generated in excel using the RANDBETWEEN([LOW], [HIGH]) function. The low and high bounds for the function are discussed below:

* 1. Team Size: The range for the low and high is between 2 and 15.
  2. Sprint Length: Range for the low is 1 weeks and high is 4 weeks.
  3. Story Point Estimation: Range for minimum story points per sprint is 40 while the maximum is 570.

According to Mark Levison, whenever a new person is added to the team, he/she increases the productivity but that also increases the communications overhead. The formula for communications within a team there are n\*(n-1)/2 connections (2016). This dictates the recommended range for the size of the team. The lower bound for the team size is two as a team cannot be smaller than two people. A single person cannot constitute a team, but the determination of the upper limit is not that easy as a team can be made as large as the team manager wants it to be. But there is the issue that it can lead to a highly inefficient team as the communication overhead is very large. For the calculation of purpose of the higher limit, the recommended team size is multiplied by a factor of two which comes out to be fifteen people in a team. The recommended team size is discussed in the next section.

Sprint length is another important factor that affects a sprint success and a sprint has multiple steps like the backlog management, sprint planning, story point estimation etc. Thus, anything smaller than one week will be too short to create a meaningful sprint. “Originally, Scrum called for one-month sprints, but nowadays many teams have been successful with two-week or even one-week sprints” (Mitch Lacey and Associates, n.d.). So, the lower limit for the sprint length is one week which is suitable for small projects. Whereas, two-week sprints are most common for software development projects. Some projects like hardware naturally take longer to complete and thus require longer sprints. Four-week sprints are more suitable for these projects. Anything longer than four-week sprint will deprive the team of the checkpoints that are required to have a successful sprint (Orgler, 2018). Hence, four weeks is chosen as the upper limit for the range.

Story point estimation range depends on the crucial assumption that the upper baseline for the story point is twenty. The upper baseline is defined as the story point limit for one user story. The recommended metric to define the upper limit is that no user story is more than sixteen hours of work (Radigan, n.d.). Based on that assumption if twenty story points is upper limit then forty can be the minimum story points in a sprint for a team of seven people. In a similar fashion, five hundred and seventy story points is the maximum story points that a team of seven developers should take on for a two-week sprint. This gives us the range of forty to five hundred and seventy.

1. MODEL PARAMTERS BOOLEAN SPECIFICATION

The model parameters are assigned Booleans to determine whether a specific parameter would contribute to the success of a sprint. The range of success for team size is three to seven after much research into the industry standard practices. A sprint of length of two to three weeks is considered as a part of the success range. For the story point estimation, the middle fifty percent is selected as the success range.

1. MODEL CONSTRUCTION, VERIFICATION, AND VALIDATION

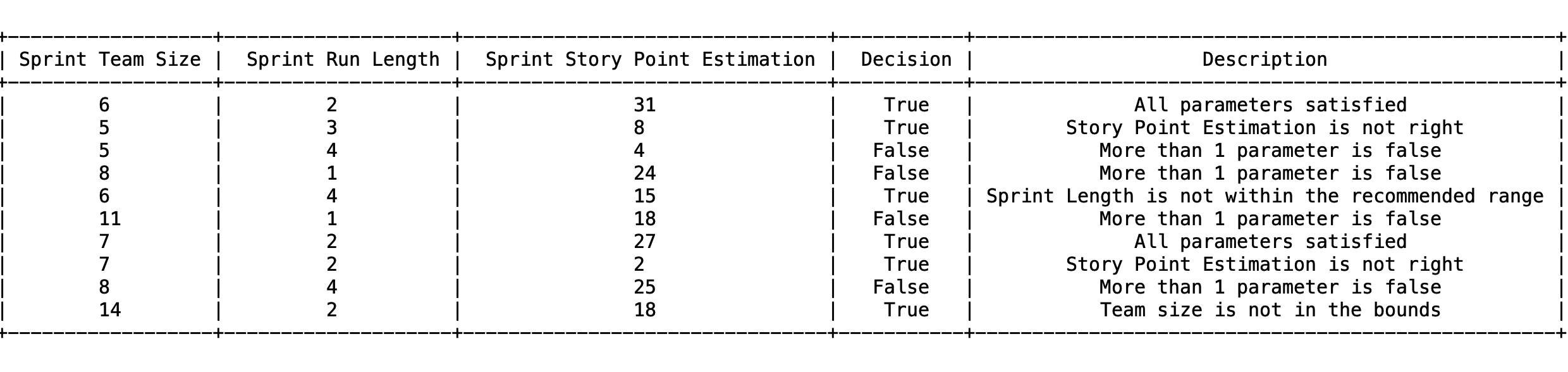
The model is constructed as a discrete event model in SimPy. The model takes a CSV file as an input for the simulation and returns Boolean output for whether a certain sprint will be successful or not. The model equation and parameters are derived from articles and research papers. Author credibility is taken into consideration when the equations and parameters are decided. Please refer to the resources on the last page for further reference and credibility verification. The model equation is verified by referring to credible sources and it is validated by creating a small sample data set and manually checking the results of the model execution. The model can be further verified and validated with the availability of real-world sprint run data. From several sprints, the data can be extracted and compiled into a CSV which can passed to the simulation. The project owner or the scrum master can be asked to add annotations for the sprints and decide whether the sprint was successful or not. The project owner’s answers can be cross checked with the simulation output to see whether the model is accurate.

1. MODEL DESIGN

The model is constructed in python using SimPy library. The model executes by looping through the entities passed in from the CSV and passing the parameter values to the environment process. The environment runs and the output is printed on the console. The input schema for the sprint data consists of three columns. The simulation code is appended in the Appendix. The code can also be accessed online on GitHub at [Sprint Success Prediction Model GitHub](https://github.com/nainkunal933/sprint-success-prediction). Sample input data is shown below along with the result output.

A close up of a logo

Description automatically generated



\

1. CONCLUSION

The agile sprint success prediction model is an attempt to create a computer model to predict the chances of success of a sprint. There are several research papers that try to achieve the same outcome using analytics driven testing, regression modelling etc., but it is the one of the first paper to tackle the problem from a simulation point of view. The model constructs a basic framework for further research and development of the model. One of the biggest challenges of the project is the availability of data. For further improvement, real life data should be used. It is also discovered that elicited data leads to false output from the model execution. With the availability of data this simulation can be verified, and validated for further development and use.

**RESORUCES**

Choosing the Team Size in Scrum. (2016, October 10). Retrieved from

<https://agilepainrelief.com/notesfromatooluser/2016/10/choosing-the-team-size-in-scrum.html>

Determining Sprint Length. (n.d.). Retrieved from

<https://www.mitchlacey.com/blog/determining-sprint-length>

Orgler, M. (2018, November 12). What is the optimal sprint length in Scrum? Retrieved from

<https://hackernoon.com/what-is-the-optimal-sprint-length-in-scrum-368e966f3243>

Atlassian. (n.d.). Secrets to agile estimation and story points. Retrieved from

<https://www.atlassian.com/agile/project-management/estimation>

**APPENDIX**

1. Simpy Code

*import* simpy  
*import* csv  
*from* prettytable *import* PrettyTable  
  
team\_size = []  
sprint\_length = []  
story\_point\_estimation = []  
  
*with open*('Simulation\_Test\_Data - Sheet1.csv') *as* csv\_file:  
 csv\_reader = csv.reader(csv\_file, delimiter=',')  
 line\_count = 0  
 *for* i *in* csv\_reader:  
 *if* line\_count == 0:  
 line\_count += 1  
 *else*:  
 team\_size.append(*int*(i[0]))  
 sprint\_length.append(*int*(i[1]))  
 story\_point\_estimation.append(*int*(i[2]))  
 line\_count += 1  
  
  
env = simpy.Environment()  
no\_of\_entity = *len*(team\_size)  
  
  
*def* sprint\_run(team\_size, length, story\_point\_estimation):  
 *# Team Size Parameter Decision  
 if* team\_size >= 5 *and* team\_size <= 7:  
 team\_size\_bool = *True  
 else*:  
 team\_size\_bool = *False  
 # Sprint Length Parameter Decision  
 if* length == 2 *or* length == 3:  
 sprint\_length\_bool = *True  
 else*:  
 sprint\_length\_bool = *False  
 # Story Point Parameter Decision  
 if* story\_point\_estimation <= 40 *and* story\_point\_estimation >= 10:  
 story\_point\_estimation\_bool = *True  
 else*:  
 story\_point\_estimation\_bool = *False  
 yield* env.timeout(0)  
 *# Final Decision  
 if* team\_size\_bool == *True and* sprint\_length\_bool == *True and* story\_point\_estimation\_bool == *True*:  
 p.add\_row([team\_size, length, story\_point\_estimation, "True", "All parameters satisfied"])  
 *return True  
 elif* team\_size\_bool == *True and* sprint\_length\_bool == *True and* story\_point\_estimation\_bool == *False*:  
 p.add\_row([team\_size, length, story\_point\_estimation, "True", "Story Point Estimation is not right"])  
 *return True  
 elif* team\_size\_bool == *True and* sprint\_length\_bool == *False and* story\_point\_estimation\_bool == *True*:  
 p.add\_row([team\_size, length, story\_point\_estimation, "True", "Sprint Length is not within the recommended range"])  
 *return True  
 elif* team\_size\_bool == *False and* sprint\_length\_bool == *True and* story\_point\_estimation\_bool == *True*:  
 p.add\_row([team\_size, length, story\_point\_estimation, "True", "Team size is not in the bounds"])  
 *return True  
 else*:  
 p.add\_row([team\_size, length, story\_point\_estimation, "False", "More than 1 parameter is false"])  
 *return False*p = PrettyTable(["Sprint Team Size", " Sprint Run Length", " Sprint Story Point Estimation", " Decision", "Description"])  
  
*# We are considering 1 sprint equal to 1 entity  
# Run the simulation  
for* i *in range*(no\_of\_entity):  
 sprint\_run\_team\_size = team\_size[i]  
 sprint\_run\_length = sprint\_length[i]  
 sprint\_run\_story\_point\_estimation = story\_point\_estimation[i]  
 env.process(sprint\_run(sprint\_run\_team\_size, sprint\_run\_length, sprint\_run\_story\_point\_estimation))  
  
output = env.run()  
*print*(p)  
  
data = p.get\_string()  
  
*with open*('console\_output.txt', 'w+') *as* f:  
 f.write(data)  
  
f.close()