



Probabilistic project sizing using Randomized Branch Sampling (RBS)

Dimitar Bakardzhiev

Managing Director
Taller Technologies Bulgaria
@dimiterbak



@dimiterbak

Dimitar Bakardzhiev is the Managing Director of Taller Technologies Bulgaria and an expert in driving successful and cost-effective technology development. As a Lean-Kanban University (LKU)-Accredited Kanban Trainer (AKT) and avid, expert Kanban practitioner, Dimitar puts lean principles to work every day when managing complex software projects with a special focus on building innovative, powerful mobile CRM solutions. Dimitar has been one of the leading proponents and evangelists of Kanban in his native Bulgaria and has published David Anderson's Kanban book as well as books by Eli Goldratt and W. Edwards Deming in the local language.

How big is our project?





Project sizing is different from delivery time estimation. Sizing estimates the probable **size** of a piece of software while delivery time estimation forecasts the **time** needed to build it. The relationship between the size of a piece of software and the time needed to deliver it is referred to as **productivity**.

Agile sizing techniques

measure User Stories

T-Shirt sizes (Small, Medium, Large and so on)

<http://www.mountangoatsoftware.com/blog/estimating-with-tee-shirt-sizes>



Story points (Fibonacci numbers or an Exponential scale)

<http://www.mountangoatsoftware.com/blog/dont-equate-story-points-to-hours>



Story points are about effort (Cohn).



But...software sizing is different from software effort estimation!

<http://www.mountangoatsoftware.com/blog/story-points-are-still-about-effort>



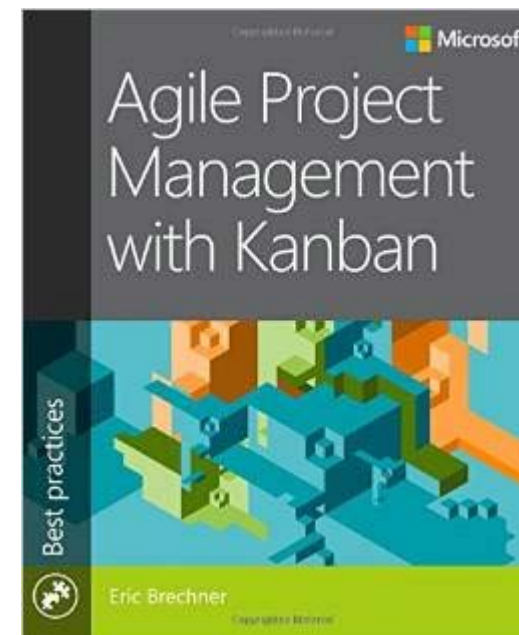
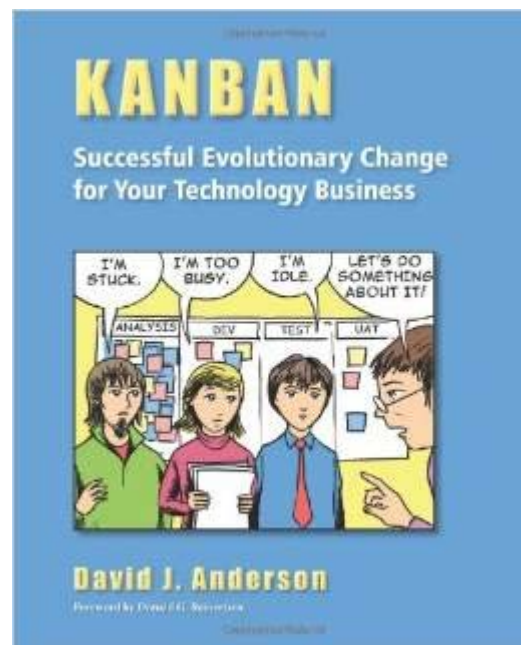
Sizing estimates the **probable size** of a piece of software while effort estimation estimates the **effort** needed to build it.



Kanban project sizing? Count!

Number of user stories,
features, use cases

Number of tasks



Any type of work items suitable for the development organization.



Example of project sizing in tasks

- We have identified 16 epics in our project
- We have identified that those 16 epics contain 102 **user stories** in total
- We have analyzed and sized every single one of those 102 user stories and arrived at a total number of 396 **tasks** for our project

This practice is time consuming and probably great part of this effort will be a pure waste!



How can we estimate the total number of stories or tasks for a project without prior identification, analysis and sizing of every single user story?



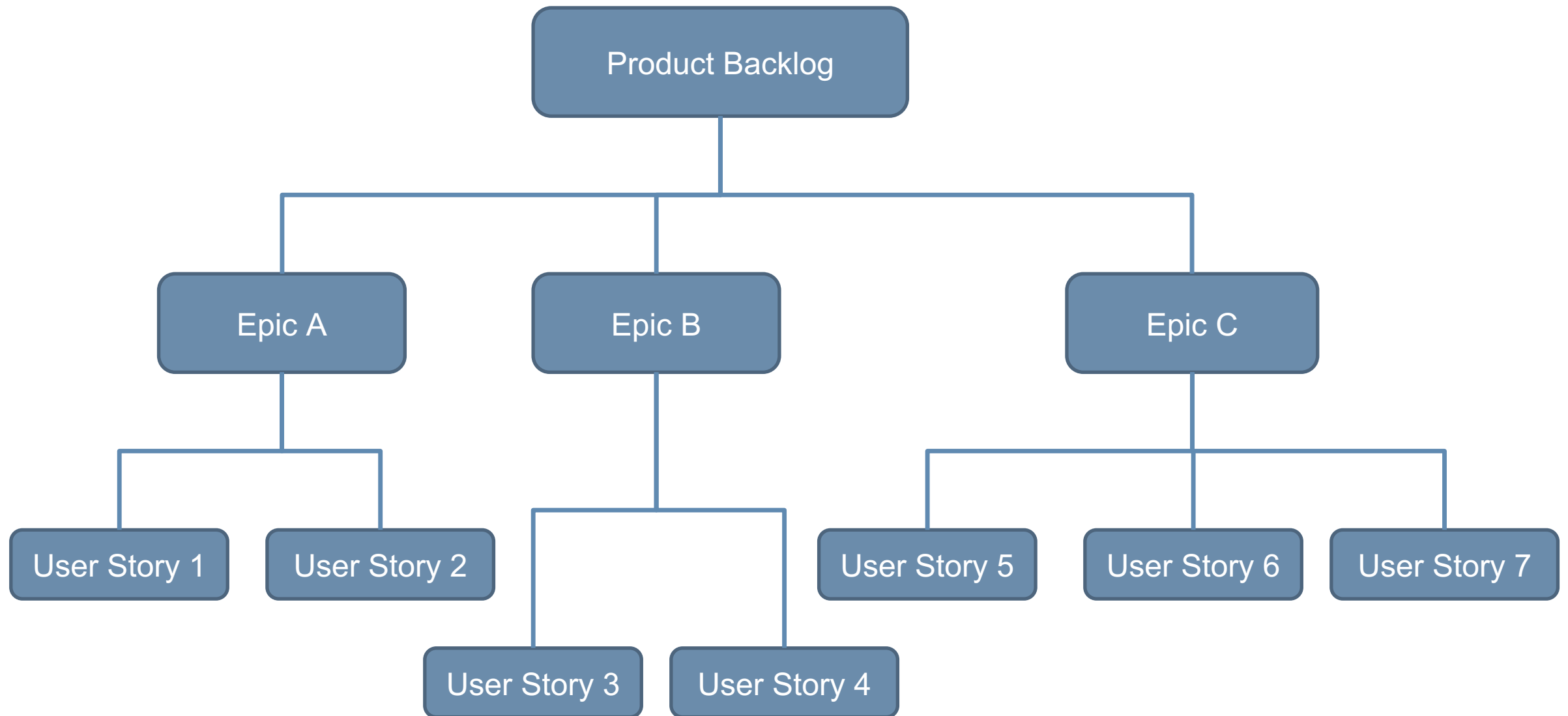
Randomized Branch Sampling



Raymond J. Jessen
1910–2003

RBS is a multi-stage unequal probability sampling method. The technique was designed to efficiently estimate the total number of fruit found in the canopy of a tree while only having to count the fruit on select branches.

Product backlog as a branching system





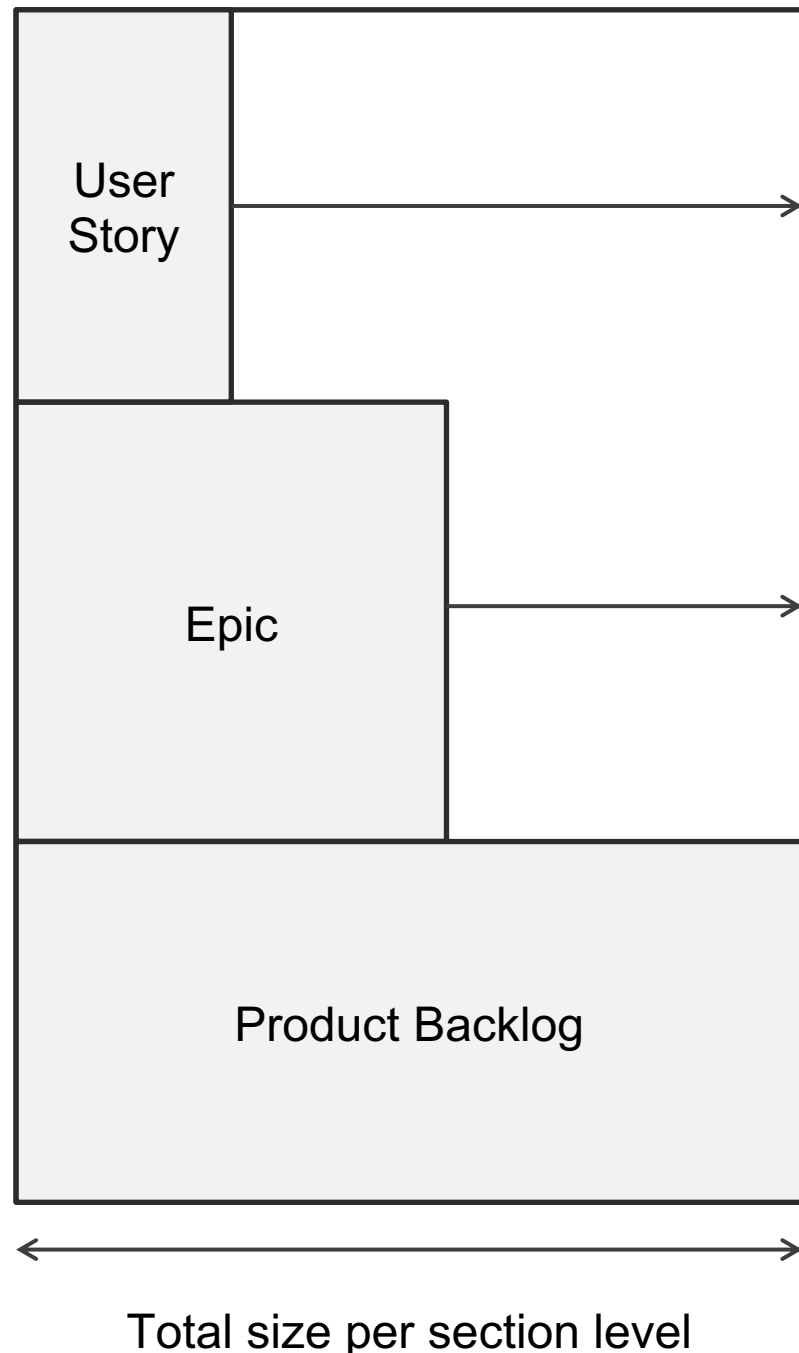
Horvitz-Thompson estimator

$$\hat{X} = \frac{x_i}{Q_i}$$

Unconditional selection probability (Q_i)

$$Q_i = \prod_{k=1}^i q_k = q_{product\ backlog} q_{epic} q_{story} = q_{epic} q_{story}$$

Application of RBS to project sizing



The user story rectangle represents the estimated size of a randomly sampled user story. The size of that user story is expanded to an estimated total project size by dividing that size by its selection probabilities which is indicated here by the arrows.

Is RBS applicable to software development?

The assumption behind using RBS for software development is that project size depends on the context – the customer, the people developing the product and the methodology they use for managing the requirements. It doesn't matter what the methodology is. What is important is the methodology to be cohesive, explicit and to be consistently applied during project execution when we slice the requirements into user stories.



The method of analysis used to break down requirements and itemize them for development has its own degree of variability. DJA, Kanban, p.221

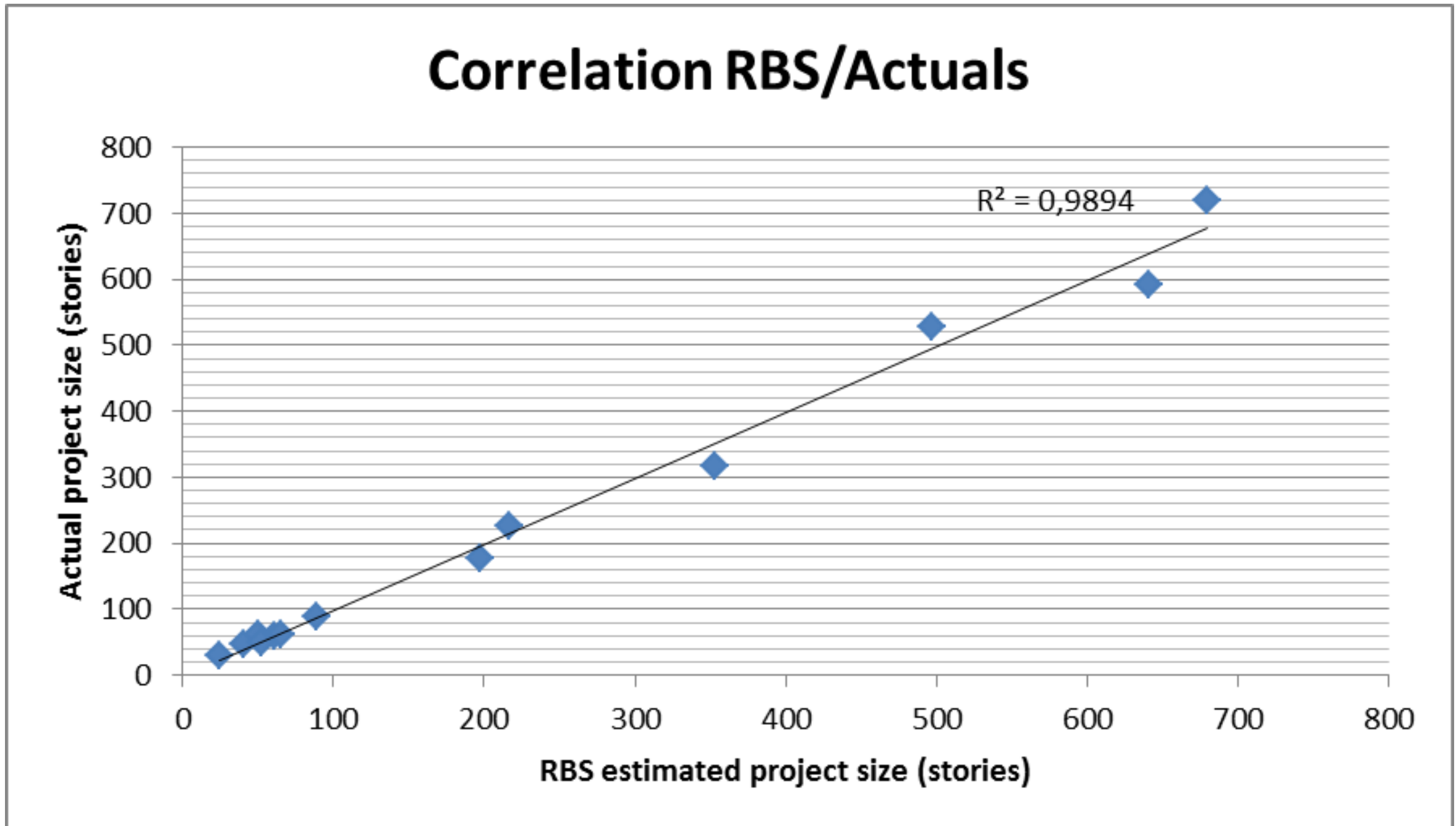


RBS compared to the actual results of 13 real ScrumDo.com projects

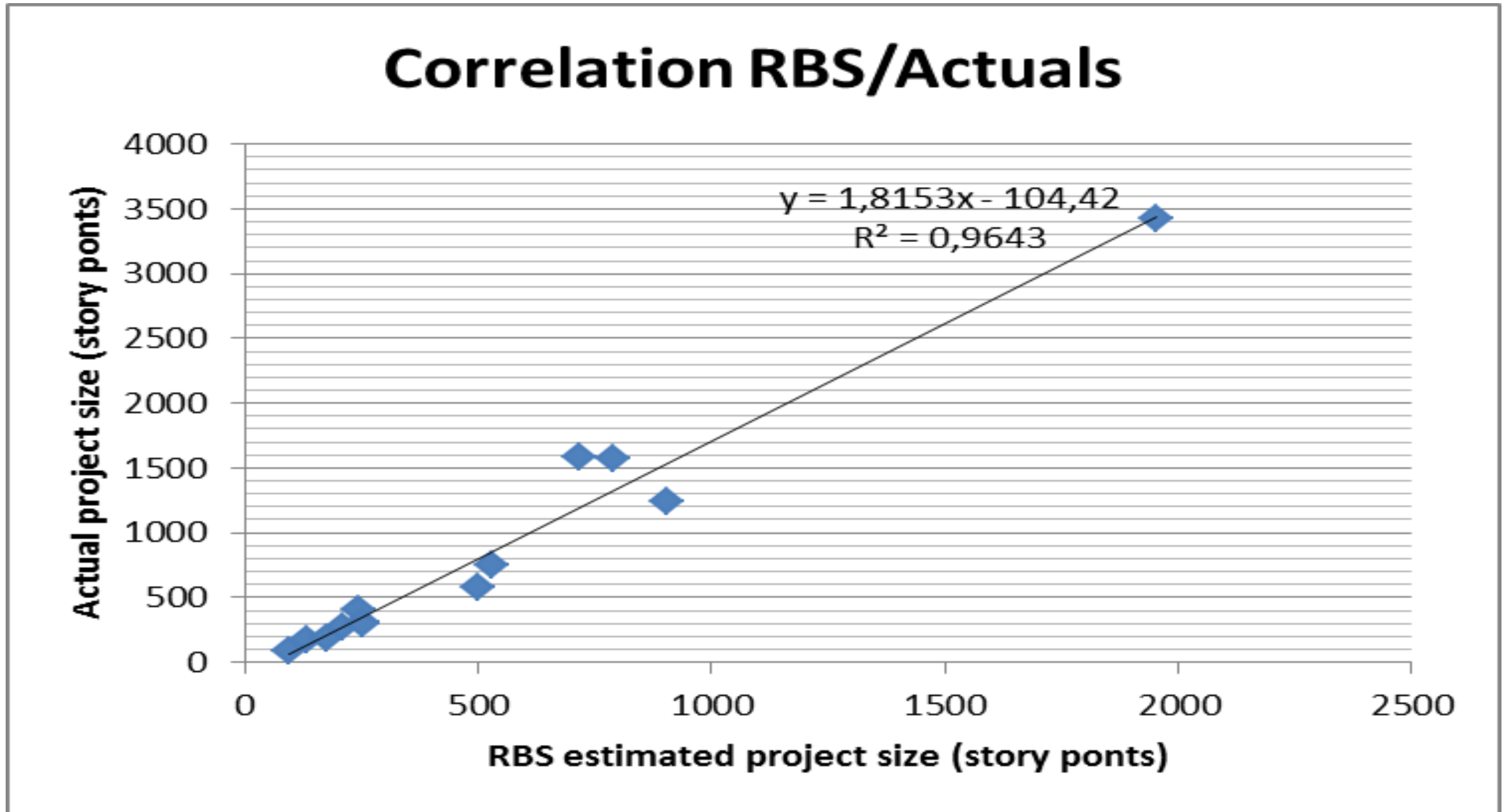
- Epic-Story-Task breakdowns
- Successful release history
- Stable teams (systems)
- Have an active ScrumDo coach or scrum master
- Commercial projects
- Have a minimum size of 12 epics/features.



RBS estimated number of stories

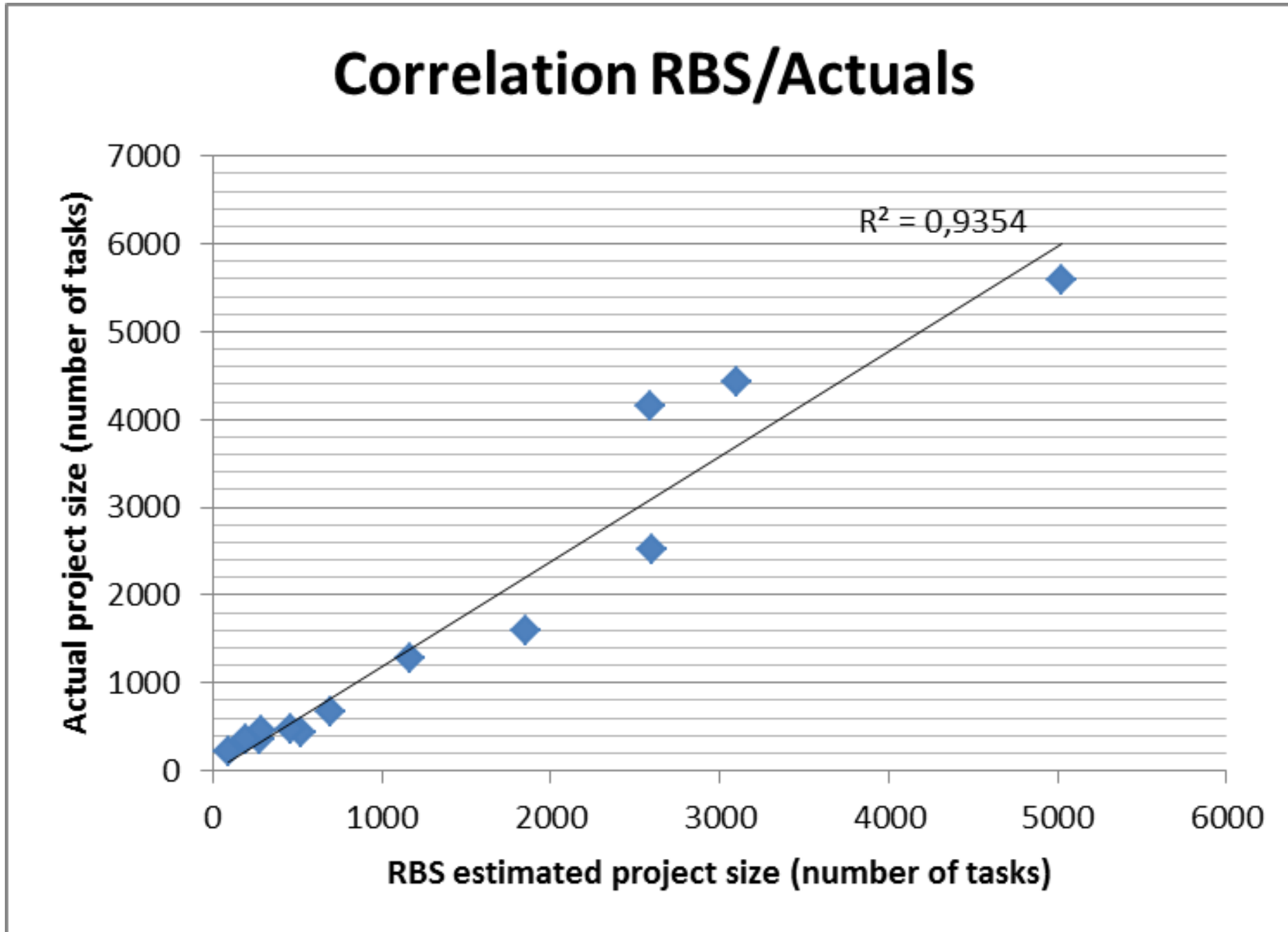


RBS estimated total story points



[ScrumDo data and results here.](#)

RBS estimated number of tasks per story





Conclusions from Scrumdo.com data

- During project execution all project teams consistently applied a methodology for slicing the requirements into user stories and tasks
- During project execution all project teams maturely managed the emergent and high-change-risk requirements
- Execution is more important than planning!!!



Applications of RBS

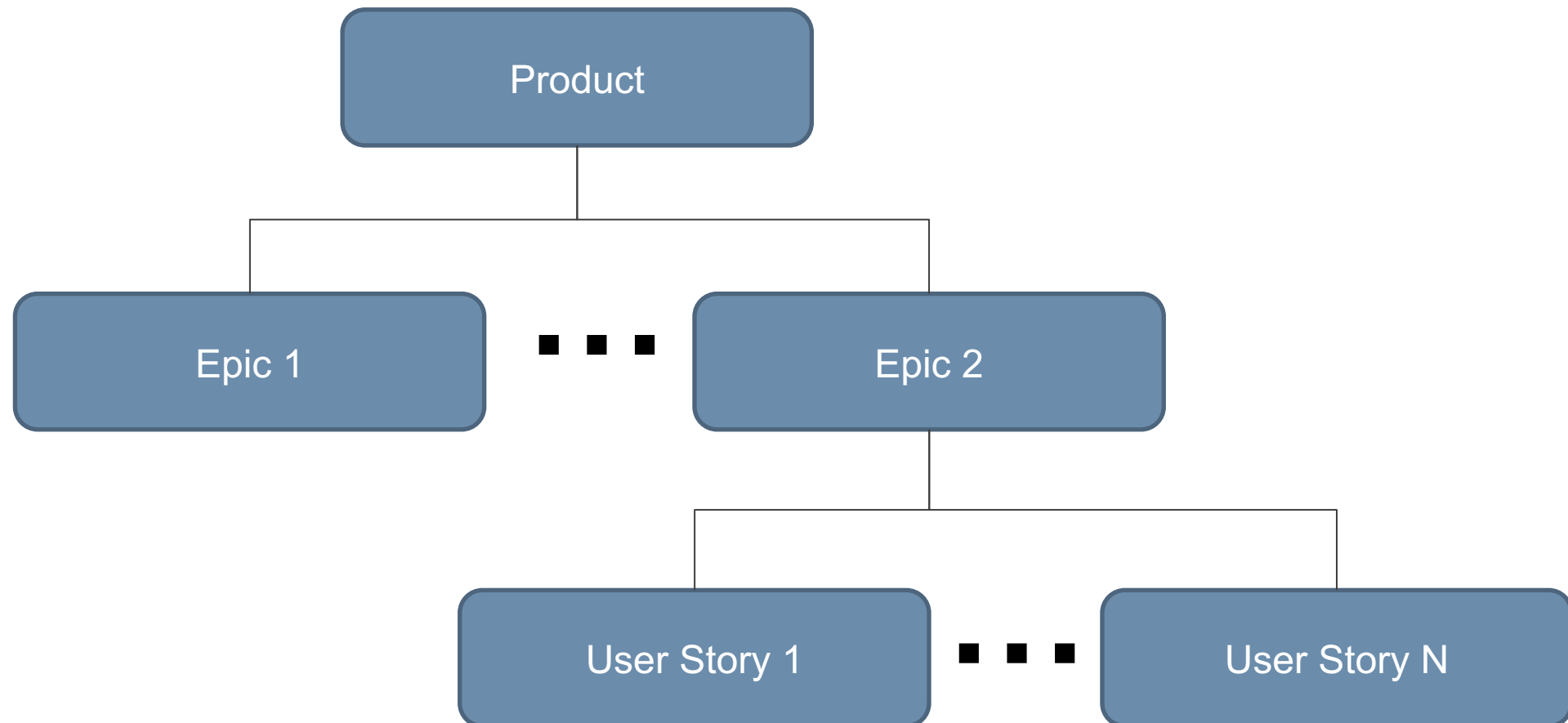
1. Applying RBS for estimating total number of user **stories** in a project
2. Applying RBS for estimating total number of **tasks** in a project
3. Applying RBS for estimating total number of BDD **scenarios** in a project
4. Assessing the **maturity** of teams in their usage of a sizing methodology

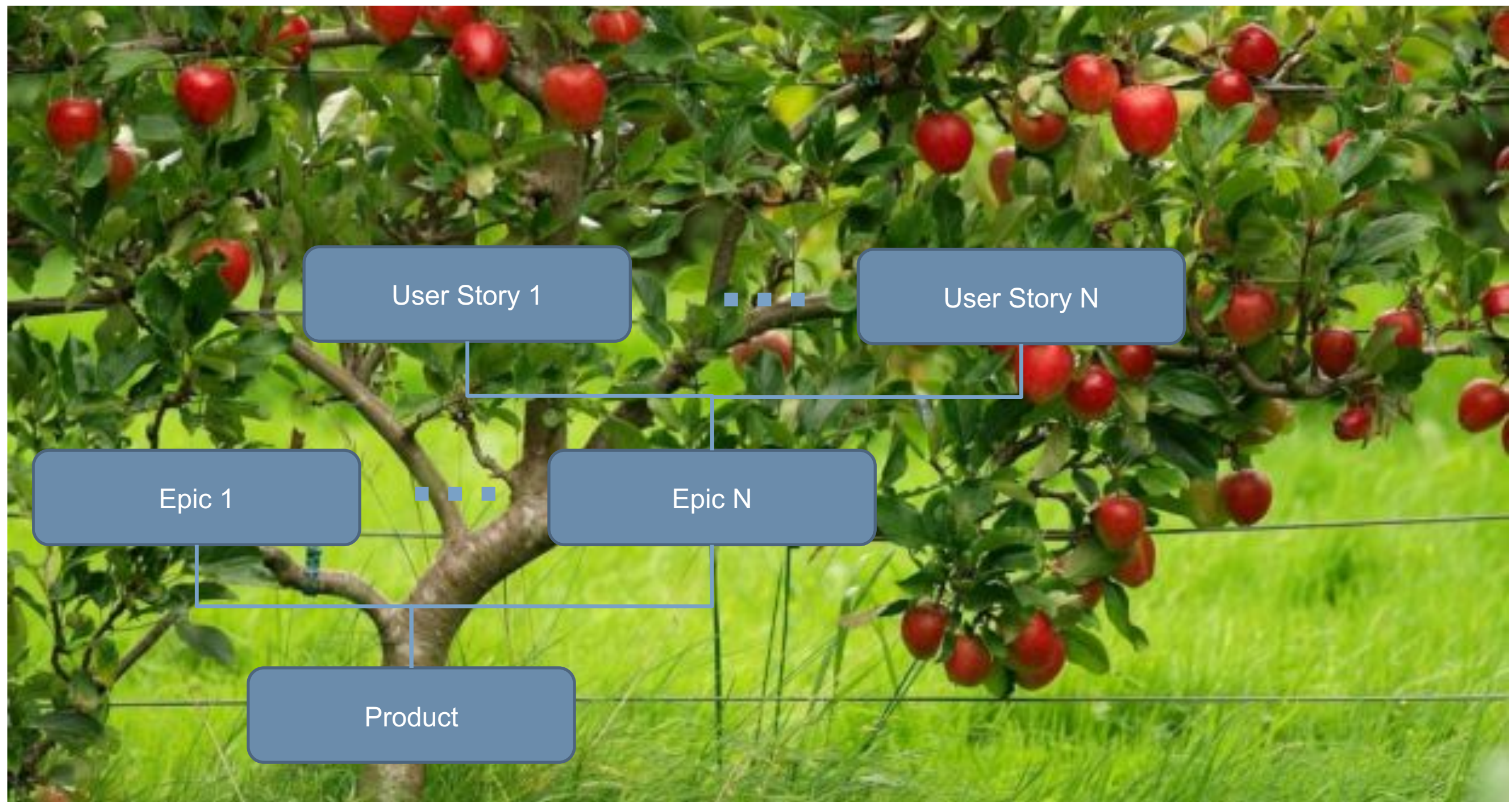


Applying RBS for estimating total number of user stories in a project



Stories based sizing model







Mapping

Product	Trunk
Epic	Branch
User Story	Terminal Shoot

RBS estimate of the of total number of user stories for a project

$$\hat{X}_i = \frac{\left(\begin{array}{c} \text{Number of user stories} \\ \text{in the sampled} \\ \text{Epic} \end{array} \right)}{\left(\frac{1}{\begin{array}{c} \text{Number of} \\ \text{Epics} \\ \text{in the project} \end{array}} \right)} \quad (1)$$

Where:

\hat{X}_i is an estimate of the total number of user stories for the project.

Total number of user stories for the project

$$\hat{X} = \frac{1}{m} \sum_{i=1}^m \hat{X}_i = \frac{1}{m} \sum_{i=1}^m \frac{\hat{S}_i}{\frac{1}{n}} \quad (2)$$

\hat{X} is an unbiased estimator of the total number of user stories for the project

\hat{S}_i is the number of user stories in the m -th epic

m is the number of estimates done

n is the number of epics in the project



Algorithm

1. Divide the project scope into epics.
2. Randomly sample one of the epics
3. Analyze how many stories are in the sampled epic.
Write down the number of stories.
4. Using formula (1) calculate one estimate of the total number of stories for the project
5. Repeat points 2-4 between 7 and 11 times
6. Using formula (2) calculate the total number of stories for the project



Following is a calculation with data from a real ScrumDo.com project.

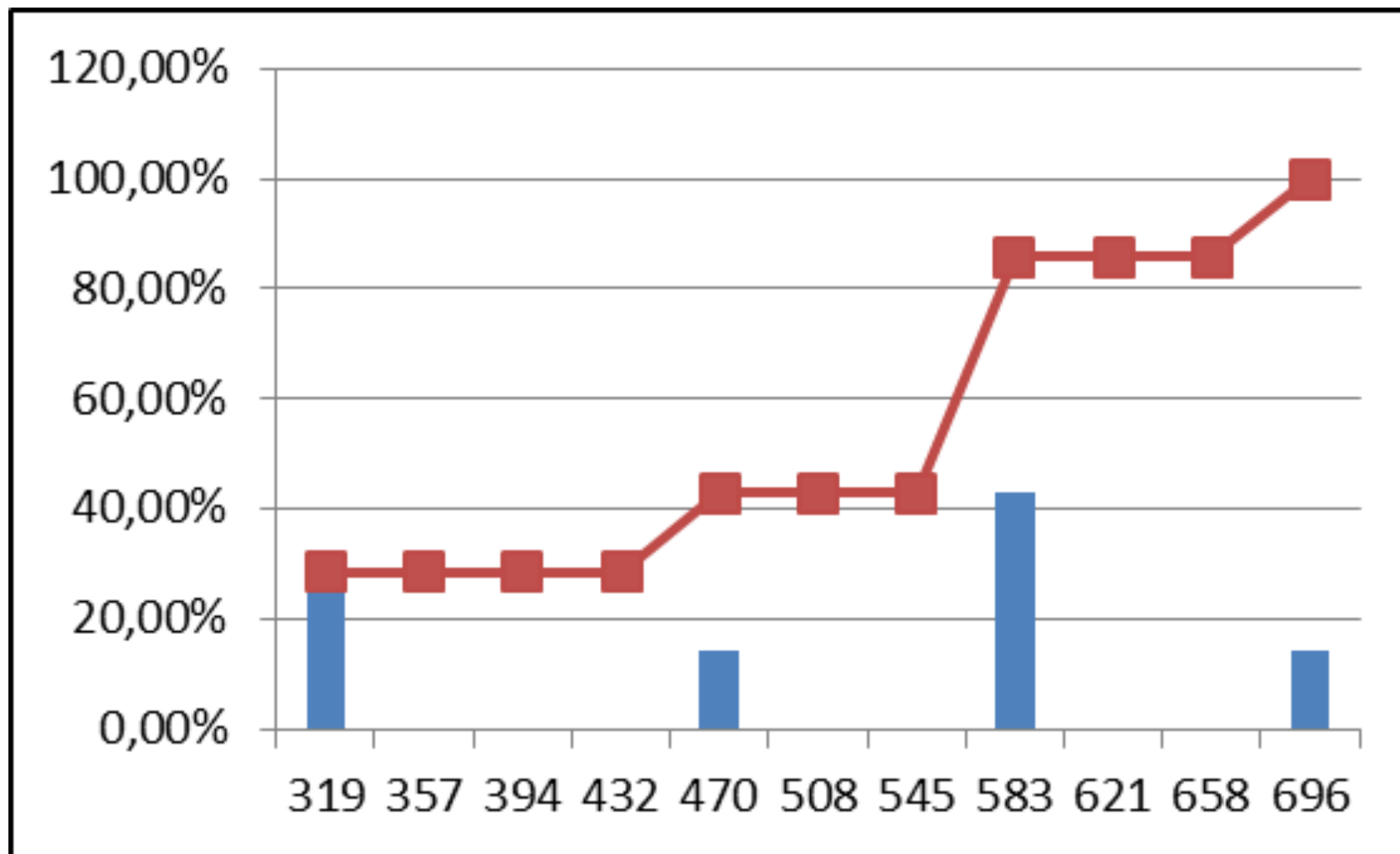
When the project finished in the backlog there were 29 epics and a total of 529 user stories.



Random epic selector	Epic #	Number of User Stories inside the epic	Epic's selection probability	Estimated total stories
0,733796	22	19	0,034483	551,00
0,596877	18	16	0,034483	464,00
0,30461	9	24	0,034483	696,00
0,988762	29	19	0,034483	551,00
0,191704	6	11	0,034483	319,00
0,184528	6	11	0,034483	319,00
0,091998	3	20	0,034483	580,00



Total Number of stories for the project



Estimated project size	497
Number of RBS paths	7
SD	53
Median	551
Mode	551

LKCE2011 - Predictability & Measurement with Kanban by David Anderson

The slide, titled "Simulating S-Curve with a Z", displays a graph with "Features" on the vertical axis (0 to 240) and "Time" on the horizontal axis (10-Feb to 20-Mar). A red S-curve is plotted, with a dashed blue line representing a linear trend. Annotations include "20%" at the start and end of the curve, "60%" at the peak, and "Slope in middle 3.5x - 5x slope at ends" with a red "5x" label. A legend at the bottom identifies stages: Inventory (blue square), Started (red square), Designed (green square), Coded (yellow square), and Complete (purple square). The slide also features the text "David J Anderson & Associates, Inc." and "Advanced Kanban".

David J Anderson
& Associates, Inc.
Advanced Kanban



DJA's way to calculate the number of work items for a project?

At 42:50 min DJA presented how to come out with the number of work items (user stories) for a project: “Randomly sample at least 7 and ideally 11 of the customer defined requirements and analyze them into work items that are meaningful to the development organization. User stories is just one example of a suitable work item type. Take the average number of the user stories per saga and multiply it with the number of sagas. The product is the number of user stories for the project.”

DJA's formula compared with RBS – they are the same!

$$\hat{S} = \frac{n}{m} \sum_{i=1}^m \hat{S}_i = \frac{1}{m} \sum_{i=1}^m \frac{\hat{S}_i}{\frac{1}{n}}$$

Where:

\hat{S} is the number of work items per saga

n is the number of sagas in the project

m is the number of sagas broken down into user stories



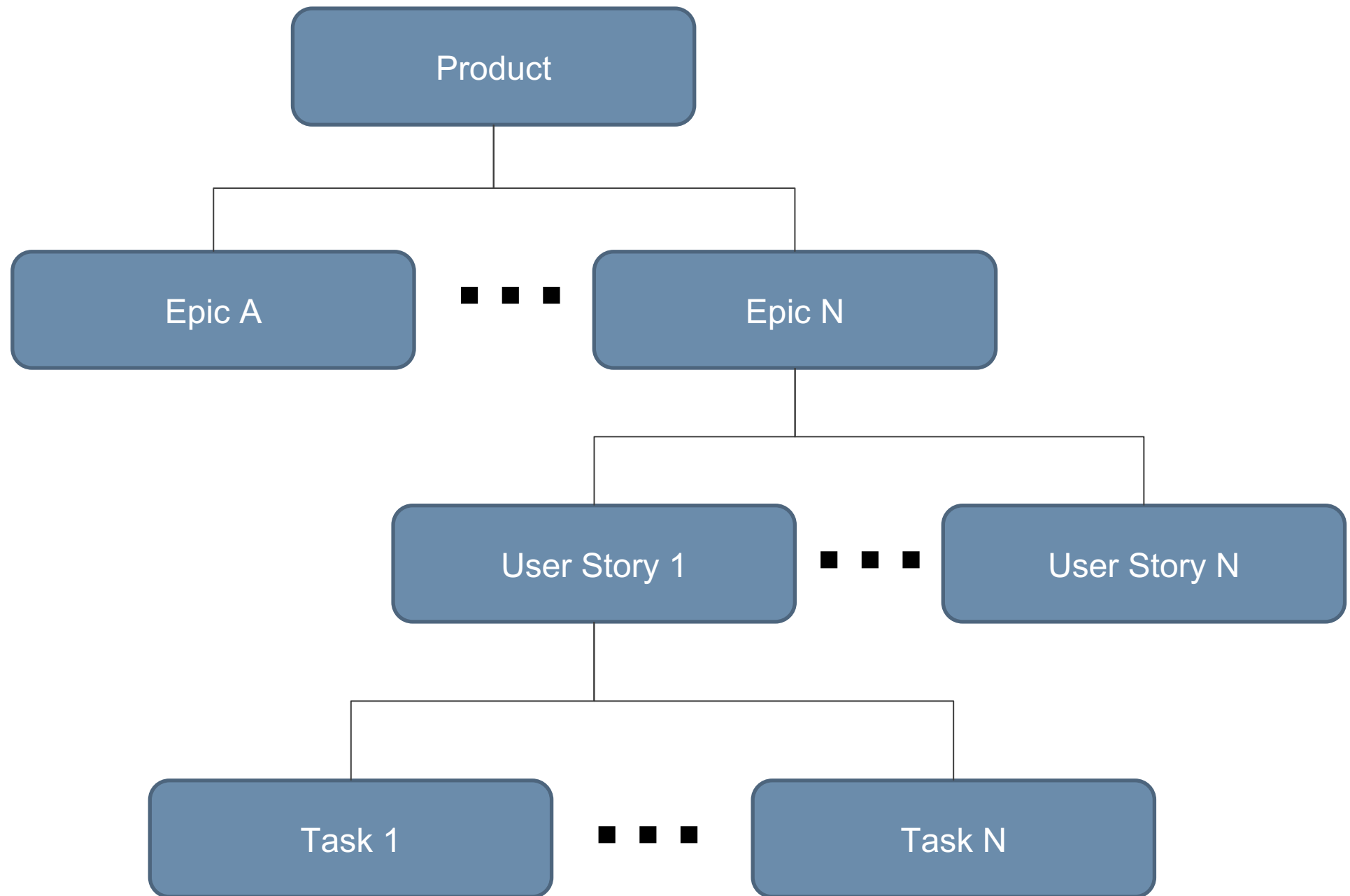
Applying RBS for estimating total number of tasks in a project

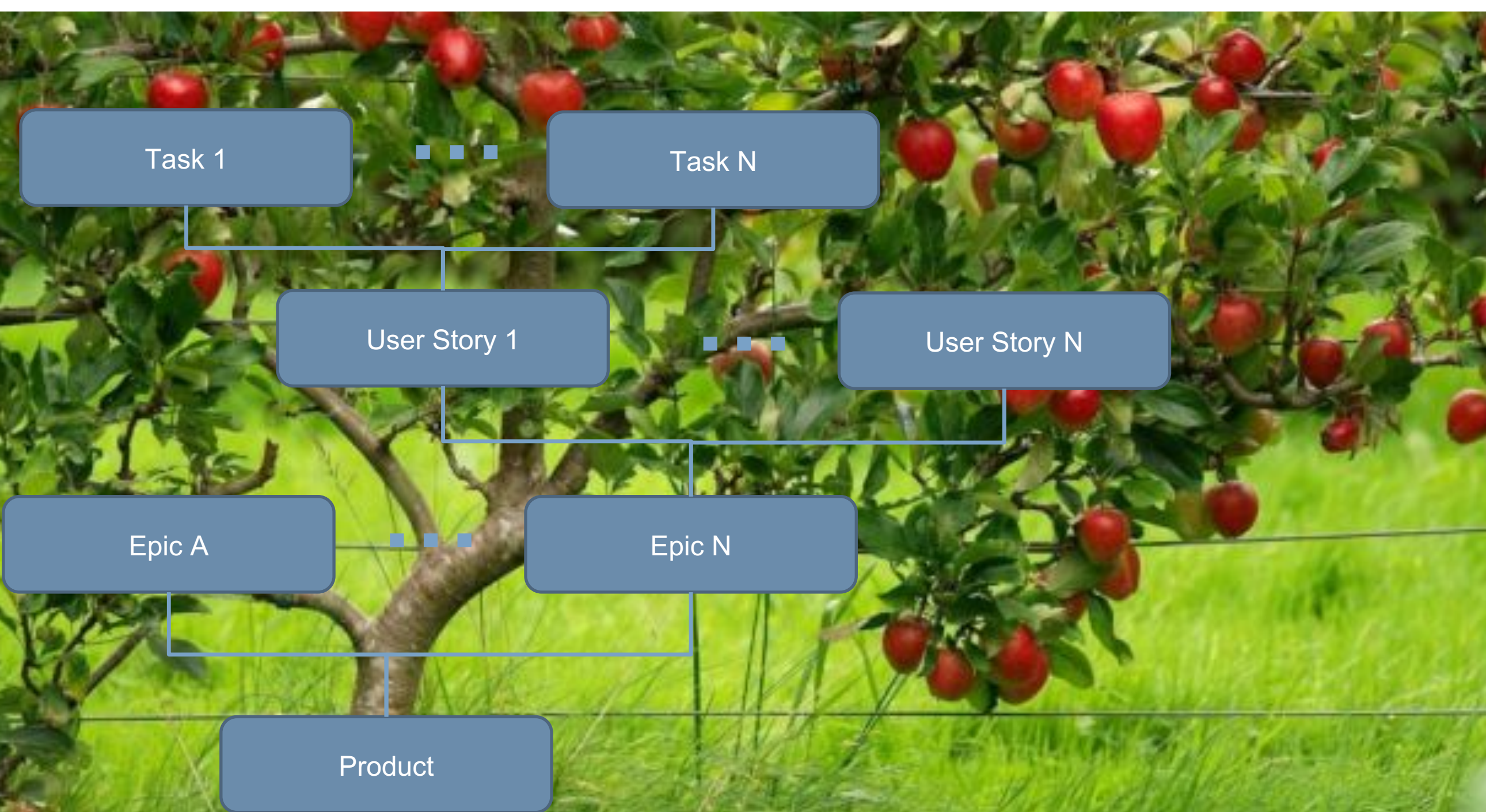


When we size each user story in the number of tasks then the project size is the total of all tasks.



Tasks based sizing model







Mapping

Product	Trunk
Epic	Branch
User Story	Terminal Shoot
Number of tasks per User story	Number of Fruit on the Shoot

Estimate of the of total number of tasks for a project

$$\hat{X}_i = \frac{\left(\begin{array}{c} \text{Number of tasks} \\ \text{in the sampled} \\ \text{User story} \end{array} \right)}{\left(\frac{1}{\left(\begin{array}{c} \text{Number of} \\ \text{Epics} \\ \text{in the project} \end{array} \right)} \right) \left(\frac{1}{\left(\begin{array}{c} \text{Number of} \\ \text{User stories} \\ \text{in the sampled Epic} \end{array} \right)} \right)} \quad (3)$$

Where:

\hat{X}_i is an unbiased estimator of the population total of the of story points for the project.



Total number of tasks for the project

$$\hat{X} = \frac{1}{m} \sum_{i=1}^m \hat{X}_i \quad (4)$$

Where:

\hat{X} is an unbiased estimator of the total number of tasks for the project.

m is the number of estimates done



Algorithm

1. Divide the project scope into epics.
2. Randomly sample one of the epics
3. Analyze how many stories are in the sampled epic.
Write down the number of stories.
4. Randomly sample one of the stories of the epic from p.2
5. Establish the tasks for the story from p.4
6. Using formula (3) calculate one estimate of the total number of tasks for the project
7. Repeat points 2-6 between 7 and 11 times
8. Using formula (4) calculate the total number of tasks for the project



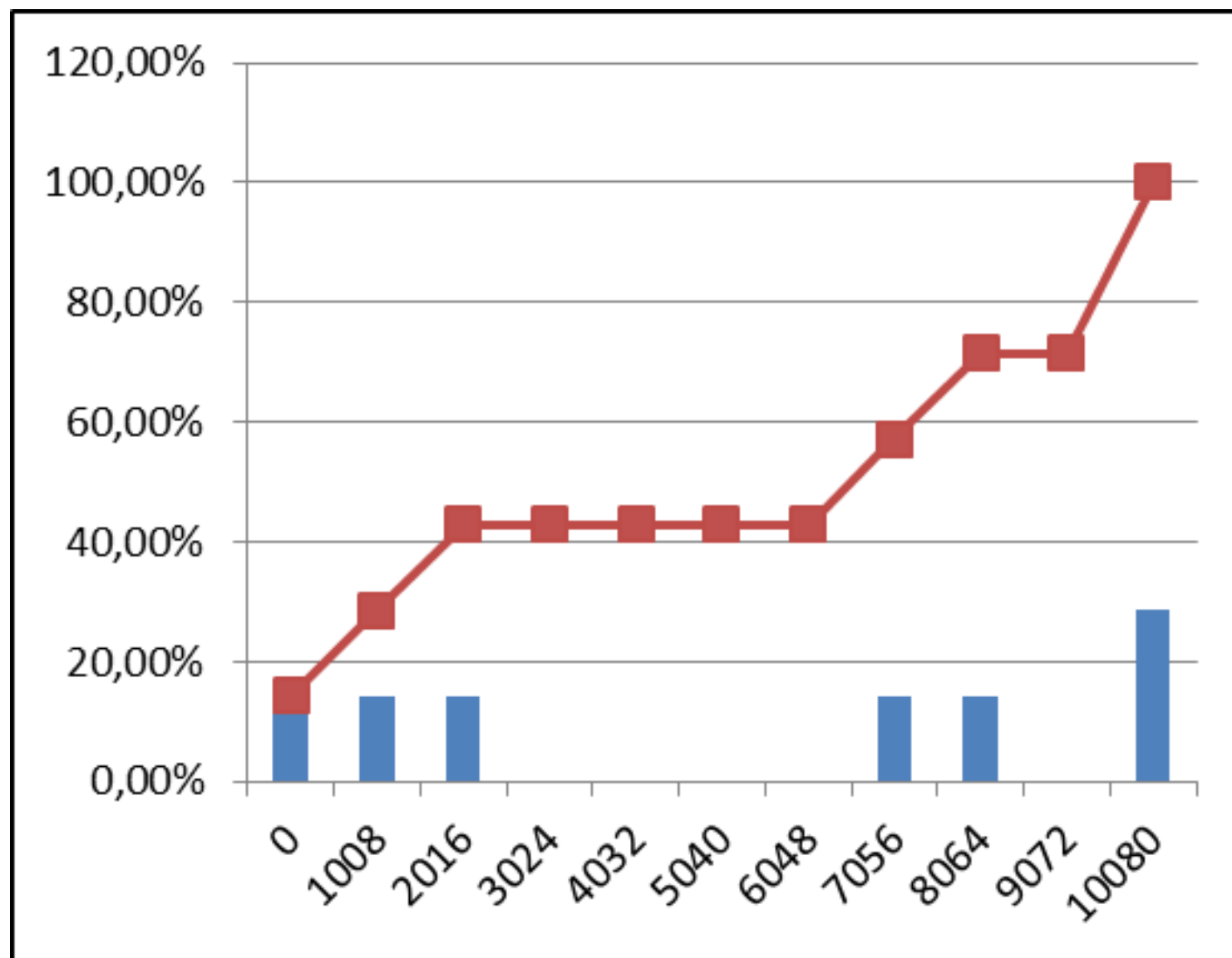
Following is a calculation with data from a real ScrumDo.com project.

When the project finished in the backlog there were 15 epics, 720 user stories and a total of 5591 tasks.



Random epic selector	Epic #	Number of User Stories inside the epic	Random story selector	Selected user story	Tasks for the selected story	Epic's selection probability	Story's selection probability	Conditional selection probability	Estimated total tasks
0,887642	14	42	0,649722871	545769	12	0,066667	0,02381	0,0015873	7560,00
0,763994	12	51	0,017087888	506420	8	0,066667	0,019608	0,0013072	6120,00
0,897303	14	42	0,571814178	541008	1	0,066667	0,02381	0,0015873	630,00
0,542088	9	37	0,559320969	544703	2	0,066667	0,027027	0,0018018	1110,00
0,510646	8	48	0,360797137	527216	14	0,066667	0,020833	0,0013889	10080,00
0,457058	7	46	0,892151817	564853	14	0,066667	0,021739	0,0014493	9660,00
0,736139	12	51	0,972991924	567925	0	0,066667	0,019608	0,0013072	0,00

Total Number of tasks for the project

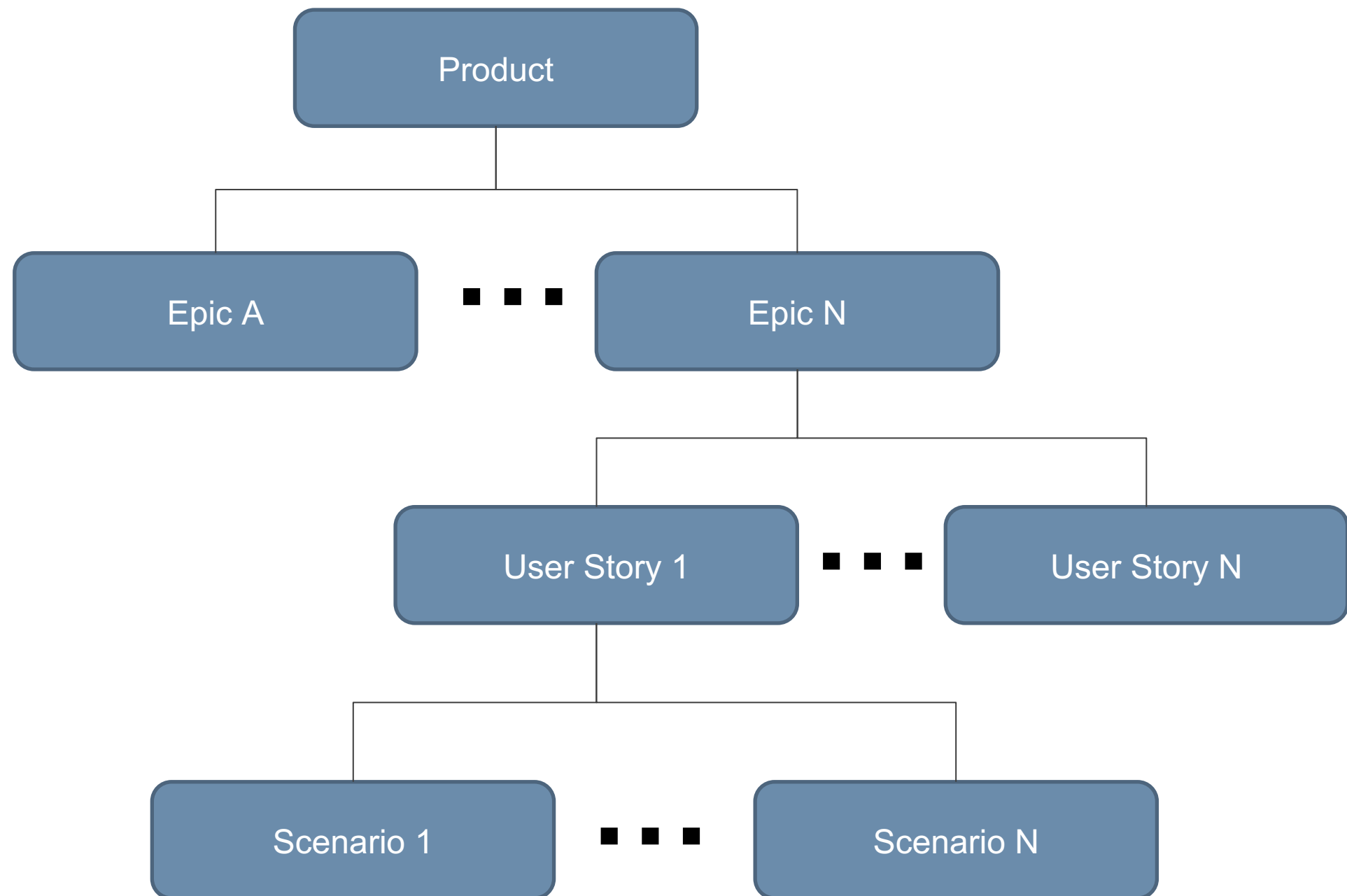


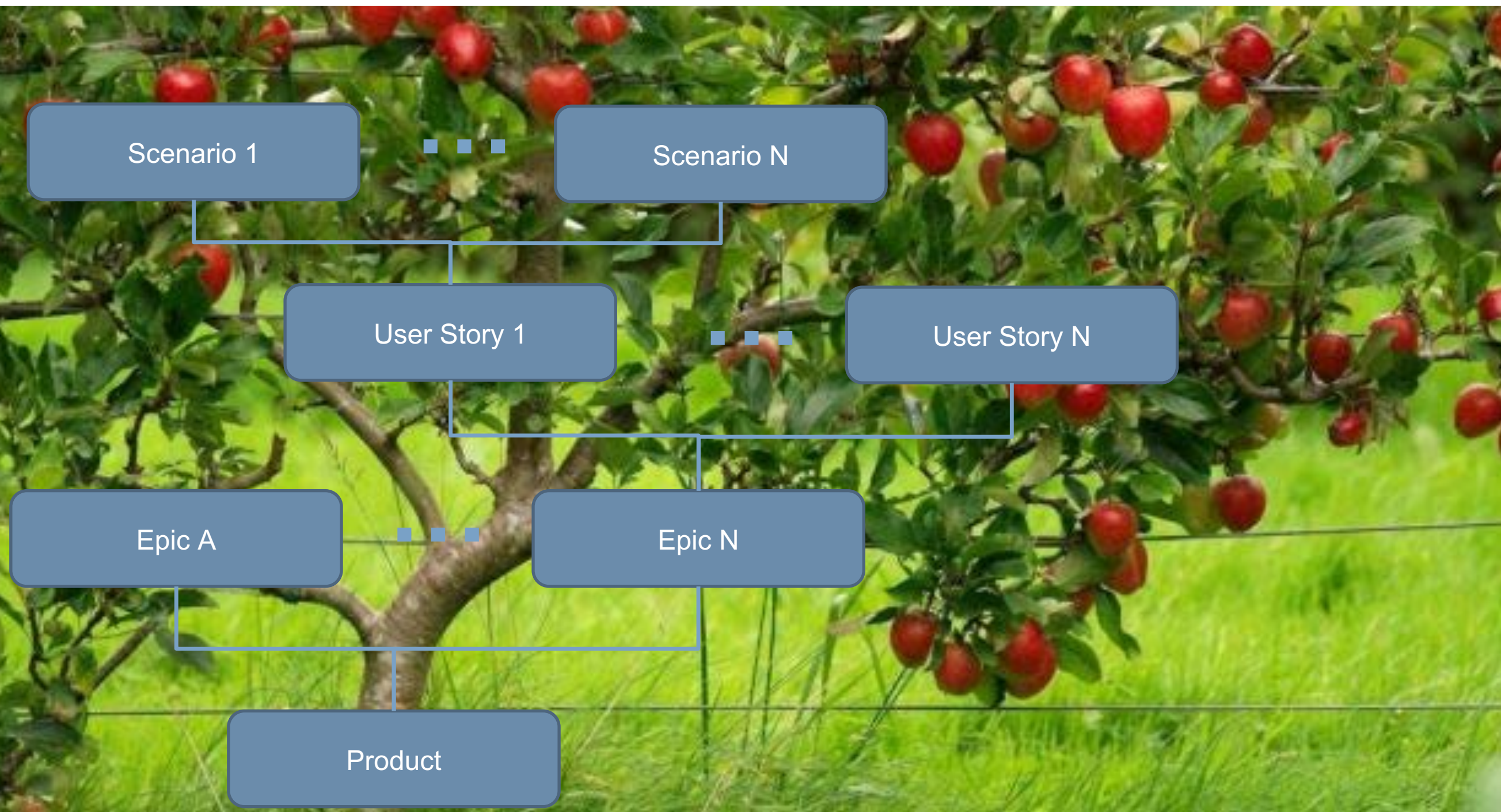
Estimated project size	5023
Number of RBS paths	7
SD	1652
Median	6120



Applying RBS for estimating total number of BDD scenarios in a project

Scenario based sizing model







What is a Scenario?

- A scenario is an acceptance test customers could understand written in their ordinary business language. It is a formal test conducted to determine whether or not the system satisfies its acceptance criteria and to enable the customer to determine whether or not to accept the system.
- A User Story can have one or more scenarios



When we size each user story in the number of scenarios then the project size is the total of all scenarios.



Mapping

Product	Trunk
Epic	Branch
User Story	Terminal Shoot
Number of Scenarios per User story	Number of Fruit on the Shoot

An estimate of the of total number of scenarios for a project

$$\hat{X}_i = \frac{\left(\begin{array}{c} \text{Number of scenarios} \\ \text{in the sampled} \\ \text{User story} \end{array} \right)}{\left(\frac{1}{\left(\begin{array}{c} \text{Number of} \\ \text{Epics} \\ \text{in the project} \end{array} \right)} \right) \left(\frac{1}{\left(\begin{array}{c} \text{Number of} \\ \text{User stories} \\ \text{in the sampled Epic} \end{array} \right)} \right)} \quad (5)$$

Where:

\hat{X}_i is an estimate of the total number of scenarios for the project.

Total number of scenarios for the project

$$\hat{X} = \frac{1}{m} \sum_{i=1}^m \hat{X}_i \quad (6)$$

Where:

\hat{X} is an unbiased estimator of the total number of scenarios for the project.

m is the number of estimates done



Algorithm

1. Divide the project scope into epics.
2. Randomly sample one of the epics
3. Analyze how many stories are in the sampled epic.
Write down the number of stories.
4. Randomly sample one of the stories of the epic from p.2
5. Establish the scenarios for the story from p.4
6. Using formula (5) calculate an estimate of the total number of scenarios for the project
7. Repeat points 2-6 between 7 and 11 times
8. Using formula (6) calculate the total number of scenarios for the project



Conclusion

- RBS is a forecasting technique for sizing software projects without prior identification, analysis and sizing of every single user story. Project size may be measured in story points, number of tasks, BDD scenarios etc.
- By running RBS on past data from actual projects, we found that the RBS would have estimated the same size without all the usual effort.
- RBS helps us to reduce uncertainty regarding “how much” software needs to be developed when we have to make portfolio related decisions, provide quotations on prospect projects etc.