

Agile – Sizing and Velocity

For any project, the question that always comes from management is, "When will it be done?"

The "waterfall" process attempts to answer this by imagining every possible task, ordering the tasks based on the critical path, and placing beginning and end dates on each task along with who is responsible for that task. An entire book could be written on problems encountered with this methodology and that is not the focus of this document.

The Agile methodology takes a very different approach. High level components and/or functionality are captured as Epic Stories (Epics). Epics generally encompass a lot of work which is only implicitly identified. Epics are then broken down into associated User Stories (Stories; known as Issues in Jira.) Stories explicitly break down the implicit work contained in the Epics into separate, actionable and deliverable chunks of work. Stories only seek to define what is required and how to know when it is done. Stories define neither how to do the work, nor when to do the work.

Example

Project: Detail Vehicle

Epic 1: Clean Exterior

Story 1 – Wash

Story 2 – Dry

Story 3 – Paint Claying

Story 4 – Polish

Story 5 – Seal/Wax

Epic 2: Clean Interior

Story 6 – Vacuum

Story 7 – Scrub and Brush Carpets

Story 8 – Clean glass

Story 9 – Leather trimming

Story 10 – Perfuming

Sidebar: What is really shown above are just story titles. An actual story should contain a lot of information. The full conversation about what a story is a conversation for another time.

Since Agile does not define how or when to do the work, how do we answer management's question of, "When will it be done?" This is where Sizing and Velocity enter the picture. You will need to read about and understand both before we get to answering management's question.

Sizing

In the Waterfall (sizing) methodology one will define that they need to do some task and it will get added to the project plan. Next, the person will think about what they must do to complete that task and try to figure out how long it will take to do that work. This mental exercise quite frequently looks like this: *"I mostly know how to do 'x' and that will probably take me 1 hour. However, I must get some info from someone else, so I am going to throw a couple hours in there for that conversation and waiting for their answer. I also know that there is always some gotcha, so I am just going to take the 3 hours and double it. Between meetings and everything else, let's just call it one day."* Sound familiar? Sometimes the time estimate is right.

Sometimes the task takes way less time and other times it takes much longer. Despite this inaccuracy, we rarely change the time estimates. If it took longer, then everything dependent in the plan gets pushed out. If it took less time, then I get to do some of my other work to get caught up or try to get ahead.

Agile does not care about time at this stage. It is a fact that the work is going to take however long it takes and not a minute more nor a minute less. Instead, sizing in agile is an effort to quantify the amount of work that needs to be done. Much like the waterfall method, we look at a story and ask ourselves what needs to be done, but we do not attach a time to it. Instead, we are evaluating how much work is involved. Is it a small amount of work? Is it medium amount of work? Is it a large amount of work? Some agile teams actually size their stories based on t-shirt sizes such as XS, S, M, L, XL.

This abstraction of the amount of effort or quantity of work involved is frequently the most difficult part of sizing, both for people new to agile and even many that have been at it for a while. This is a good time to make an important note on agile sizing. Because this is not a measurement of time nobody should be asking themselves how long it would take if they were doing the work. Agile sizing assumes ideal conditions and that all knowledge, skills, and tools exist to do the work.

It would take me 10 hours or more to detail my vehicle. A professional could detail my vehicle in an hour. Notice that the actual work did not change, while the time changed by an order of magnitude. It is still "one detailing" whether I do it or the professional does it. What did change is the knowledge, skills, and tools.

At this point you might be wondering how much work is "medium" and who gets to decide. The answer is that medium is any amount of work that you and the rest of the team agree is medium. Going back to the vehicle detailing, if you and your team feel that Story 1 is medium then it is medium. If Story 2 is approximately the same amount of work as Story 1 then it is also medium. If Story 2 is less work, then it is a Small or eXtra Small depending on how much less work. Similarly, if it is more work than Story 1 then will be a Large or eXtra Large instead. There are a couple of important rules to keep in mind. First, the size does not change because you acquire the knowledge, skill, and/or tools that allow you to complete the task faster.

Second, effort in this usage is not about how long the work will take nor any lack of knowledge, skill, and/or tool. Even with perfect knowledge, skills, and tools some things just require more effort than others. Sizes are always based on ideal conditions.

We need to talk about one more concept in sizing before we move on to Velocity and finally answering the manager's question. So far, we have not used any numbers when talking about sizing and this is intentional. Agile sizing can be very uncomfortable until it becomes familiar. I would even encourage people new to sizing to step away from t-shirt sizes and use colors. If one story is orange, then any other story that is similar in the quantity of work is also orange. When you encounter a story that is an appreciably different amount of effort then call it red. Repeat this until you have all your stories grouped. This process helps move one completely away from time and to start thinking about all work in relation to other work.

Now let's introduce numbers to the sizing process. It is important to note that this changes nothing about anything previously stated. The numbers are still an abstraction and almost strictly relevant to calculating velocity. One could easily translate the t-shirt sizes (xs, s, m, l, xl) to 1, 2, 3, 4, 5 and call it a day. The same could be said for the colors as long as they're ordered from smallest to largest. What many experienced teams will use instead of sequential integers is the Fibonacci sequence 1, 2, 3, 5, 8, 13, instead. These are still relative numbers. A 1 is less work than 2 while 3 is more work. The value here is in the meta-data. Teams tend to have a reasonably high precision when estimating a story as a 1, 2, or 3. What happens is that as stories get more complex our ability to accurately estimate diminishes. Something we might call a 4 could just as easily be a 5 or even a 6. For each step up the inaccuracy greatly increases. In fact, the general rule is that any story that is sized as an 8 is almost assuredly work that needs to become two or more stories and a 13 is a signal that we do not have sufficient information.

Velocity

Now that we are sizing our stories numerically, we can start tracking a metric known as velocity.

To properly explain velocity, there is another term that needs to be introduced, first. The Agile methodology says to pick a particular interval and to work on the most important stories during that interval. The interval is functionally no shorter than one work week and typically not longer than two weeks. Since we are iterating on the project work each interval, this period is sometimes called an iteration. In Scrum, which is a particular framework built on Agile principles, this interval or iteration is called a Sprint.

Each iteration the team and customers come together to identify the most important stories to be worked on. A story might be most important because it is integral to other stories. A story might be most important because it delivers urgent required functionality or other business value. The entire collection of stories that have been created are called the "Backlog". During the Sprint Planning selected stories are moved from the Backlog to the current sprint. This

process continues with a new team until the team feels there is enough work in the current sprint to keep them busy for the entire iteration.

At the end of the first sprint there will likely be some stories that have been completed, some still in-progress, and maybe some not even started. To calculate velocity, we simply add up the size of every completed story. Stories that are in-progress or not started are not counted. For discussion purposes, let's assume that the velocity for Sprint 1 is 13. The end of Sprint 1 marks the beginning of Sprint 2 and another Sprint Planning. The customers can decide that everything in progress and not started are still priorities or they may feel that priorities have changed. In any case, the goal now is to populate Sprint 2 with stories so that the combined total size is 15-20. The numerical sizes are commonly called units, story points, or just points. The idea is to have enough work to keep the team busy for the entire sprint.

At the end of Sprint 2 and every subsequent sprint the total points are calculated for that sprint. Additionally, the sprint totals for some number of prior sprints (typically 4-6) are added together and an average sprint velocity is calculated. It is this average number that becomes the official team velocity and is used to decide how much work to put into each sprint.

At long last, we can now answer management's question. However, before we do, there is one more significant difference between Agile and Waterfall. That difference is what is "it". Where Waterfall seeks to define the entire lifespan of a project, pure Agile only concerns itself with delivering the correct, customer chosen, value every iteration. Any given team typically works on the same project from one iteration to the next, but there is no rule saying this is the case. As the needs of the business change, so will the stories in any given sprint. In Agile, "it" is whatever the business needs it to be at a given point in time.

In not so pure Agile, the desire is to have an estimate of when a project will be completed. Once a team has an established velocity (average of at least four iterations with the same team composition) then this answer can be derived by sizing the stories required to complete the project, adding the size of every story together to get a total, and then dividing by the team velocity. If an iteration is 1-week, velocity is 15, and there are 227 story points then a reasonable estimate is that the project can be completed in 16 weeks. ($227/15 = 15.13$, round up) There are processes, tools, and strategies for tracking stories and creating a running set of metrics for answering this question and many others that are near and dear to managers.

Important notes about velocity:

- No two teams will necessarily have the same velocity because we almost never do our work under ideal conditions with all necessary knowledge, skill, and tools.
- Even when teams do have the same velocity, it does not mean that they did the same amount of work. Sizes are abstractions and only have meaning within the team that performed the sizing. Velocity should never be used to compare two teams. One team

may have a completely different understanding of how much work 2 story points represents. Team 1 may call a story a 2 while team 2 calls the same story a 5. In this case, both teams are doing the same amount of work.

- As a team's knowledge, skills, and available tools improve so will their velocity. It is acceptable, and necessary, for teams to have honest conversations about whether there are impediments to getting more done. This is most effectively achieved by performing regular retrospectives to examine what is working well and identifying areas for improvement which can include training, tools, elimination of blockers, team building, or anything else that holds a team back.
- Velocity in and of itself should never be the goal. Managers! Do not do this to your agile teams. If you ask for twice the velocity you will absolutely get the velocity, but you will not get any additional work delivered. Story points will simply get inflated to reach your new target. Instead, attend the retrospectives so that you can act on the needs the team identifies.
- When the team composition changes the velocity will also likely change. Any significant change will require resetting to Sprint 1 status with an entirely new calculation of average velocity after 4-6 iterations.

Closing

There are lots of questions and considerations that have not been covered by this document.

Example: How do you size a story like Story 10 in the vehicle detailing example when it requires so little effort it is not even comparable to a '1'? There is a rare '0' point story.

The goal of this document was to introduce the concepts of Size and Velocity without getting lost in various edge cases and other concepts. Please submit your questions so they can be used to improve this document, inform the creation of a next-level document, and/or become part of a FAQ.

Thank you for your time!

Larry Dorman
Senior Site Reliability Engineer
ldorman@nvidia.com
[NVIDIA](#)

Sizing

In the Waterfall (sizing) methodology one will define that they need to do some task and it will get added to the project plan. Next, the person will think about what they must do to complete that task and try to figure out how long it will take to do that work. This mental exercise quite frequently looks like this: *"I mostly know how to do 'x' and that will probably take me 1 hour. However, I must get some info from someone else, so I am going to throw a couple hours in there for that conversation and waiting for their answer. I also know that there is always some gotcha, so I am just going to take the 3 hours and double it. Between meetings and everything else, let's just call it one day."* Sound familiar? Sometimes the time estimate is right.

Sometimes the task takes way less time and other times it takes much longer. Despite this inaccuracy, we rarely change the time estimates. If it took longer, then everything dependent in the plan gets pushed out. If it took less time, then I get to do some of my other work to get caught up or try to get ahead.

Agile does not care about time at this stage. It is a fact that the work is going to take however long it takes and not a minute more nor a minute less. Instead, sizing in agile is an effort to quantify the amount of work that needs to be done. Much like the waterfall method, we look at a story and ask ourselves what needs to be done, but we do not attach a time to it. Instead, we are evaluating how much work is involved. Is it a small amount of work? Is it medium amount of work? Is it a large amount of work? Some agile teams actually size their stories based on t-shirt sizes such as XS, S, M, L, XL.

This abstraction of the amount of effort or quantity of work involved is frequently the most difficult part of sizing, both for people new to agile and even many that have been at it for a while. This is a good time to make an important note on agile sizing. Because this is not a measurement of time nobody should be asking themselves how long it would take if they were doing the work. Agile sizing assumes ideal conditions and that all knowledge, skills, and tools exist to do the work.

It would take me 10 hours or more to detail my vehicle. A professional could detail my vehicle in an hour. Notice that the actual work did not change, while the time changed by an order of magnitude. It is still "one detailing" whether I do it or the professional does it. What did change is the knowledge, skills, and tools.

At this point you might be wondering how much work is "medium" and who gets to decide. The answer is that medium is any amount of work that you and the rest of the team agree is medium. Going back to the vehicle detailing, if you and your team feel that Story 1 is medium then it is medium. If Story 2 is approximately the same amount of work as Story 1 then it is also medium. If Story 2 is less work, then it is a Small or eXtra Small depending on how much less work. Similarly, if it is more work than Story 1 then will be a Large or eXtra Large instead. There are a couple of important rules to keep in mind. First, the size does not change because you acquire the knowledge, skill, and/or tools that allow you to complete the task faster. Second, effort in this usage is not about how long the work will take nor any lack of knowledge,

skill, and/or tool. Even with perfect knowledge, skills, and tools some things just require more effort than others. Sizes are always based on ideal conditions.

We need to talk about one more concept in sizing before we move on to Velocity and finally answering the manager's question. So far, we have not used any numbers when talking about sizing and this is intentional. Agile sizing can be very uncomfortable until it becomes familiar. I would even encourage people new to sizing to step away from t-shirt sizes and use colors. If one story is orange, then any other story that is similar in the quantity of work is also orange. When you encounter a story that is an appreciably different amount of effort then call it red. Repeat this until you have all your stories grouped. This process helps move one completely away from time and to start thinking about all work in relation to other work.

Now let's introduce numbers to the sizing process. It is important to note that this changes nothing about anything previously stated. The numbers are still an abstraction and almost strictly relevant to calculating velocity. One could easily translate the t-shirt sizes (xs, s, m, l, xl) to 1, 2, 3, 4, 5 and call it a day. The same could be said for the colors as long as they're ordered from smallest to largest. What many experienced teams will use instead of sequential integers is the Fibonacci sequence 1, 2, 3, 5, 8, 13, instead. These are still relative numbers. A 1 is less work than 2 while 3 is more work. The value here is in the meta-data. Teams tend to have a reasonably high precision when estimating a story as a 1, 2, or 3. What happens is that as stories get more complex our ability to accurately estimate diminishes. Something we might call a 4 could just as easily be a 5 or even a 6. For each step up the inaccuracy greatly increases. In fact, the general rule is that any story that is sized as an 8 is almost assuredly work that needs to become two or more stories and a 13 is a signal that we do not have sufficient information.

Velocity

Now that we are sizing our stories numerically, we can start tracking a metric known as velocity.

To properly explain velocity, there is another term that needs to be introduced, first. The Agile methodology says to pick a particular interval and to work on the most important stories during that interval. The interval is functionally no shorter than one work week and typically not longer than two weeks. Since we are iterating on the project work each interval, this period is sometimes called an iteration. In Scrum, which is a particular framework built on Agile principles, this interval or iteration is called a Sprint.

Each iteration the team and customers come together to identify the most important stories to be worked on. A story might be most important because it is integral to other stories. A story might be most important because it delivers urgent required functionality or other business value. The entire collection of stories that have been created are called the "Backlog". During the Sprint Planning selected stories are moved from the Backlog to the current sprint. This

Sizing

In the Waterfall (sizing) methodology one will define that they need to do some task and it will get added to the project plan. Next, the person will think about what they must do to complete that task and try to figure out how long it will take to do that work. This mental exercise quite frequently looks like this: *"I mostly know how to do 'x' and that will probably take me 1 hour. However, I must get some info from someone else, so I am going to throw a couple hours in there for that conversation and waiting for their answer. I also know that there is always some gotcha, so I am just going to take the 3 hours and double it. Between meetings and everything else, let's just call it one day."* Sound familiar? Sometimes the time estimate is right.

Sometimes the task takes way less time and other times it takes much longer. Despite this inaccuracy, we rarely change the time estimates. If it took longer, then everything dependent in the plan gets pushed out. If it took less time, then I get to do some of my other work to get caught up or try to get ahead.

Agile does not care about time at this stage. It is a fact that the work is going to take however long it takes and not a minute more nor a minute less. Instead, sizing in agile is an effort to quantify the amount of work that needs to be done. Much like the waterfall method, we look at a story and ask ourselves what needs to be done, but we do not attach a time to it. Instead, we are evaluating how much work is involved. Is it a small amount of work? Is it medium amount of work? Is it a large amount of work? Some agile teams actually size their stories based on t-shirt sizes such as XS, S, M, L, XL.

This abstraction of the amount of effort or quantity of work involved is frequently the most difficult part of sizing, both for people new to agile and even many that have been at it for a while. This is a good time to make an important note on agile sizing. Because this is not a measurement of time nobody should be asking themselves how long it would take if they were doing the work. Agile sizing assumes ideal conditions and that all knowledge, skills, and tools exist to do the work.

It would take me 10 hours or more to detail my vehicle. A professional could detail my vehicle in an hour. Notice that the actual work did not change, while the time changed by an order of magnitude. It is still "one detailing" whether I do it or the professional does it. What did change is the knowledge, skills, and tools.

At this point you might be wondering how much work is "medium" and who gets to decide. The answer is that medium is any amount of work that you and the rest of the team agree is medium. Going back to the vehicle detailing, if you and your team feel that Story 1 is medium then it is medium. If Story 2 is approximately the same amount of work as Story 1 then it is also medium. If Story 2 is less work, then it is a Small or eXtra Small depending on how much less work. Similarly, if it is more work than Story 1 then will be a Large or eXtra Large instead. There are a couple of important rules to keep in mind. First, the size does not change because you acquire the knowledge, skill, and/or tools that allow you to complete the task faster. Second, effort in this usage is not about how long the work will take nor any lack of knowledge,

skill, and/or tool. Even with perfect knowledge, skills, and tools some things just require more effort than others. Sizes are always based on ideal conditions.

We need to talk about one more concept in sizing before we move on to Velocity and finally answering the manager's question. So far, we have not used any numbers when talking about sizing and this is intentional. Agile sizing can be very uncomfortable until it becomes familiar. I would even encourage people new to sizing to step away from t-shirt sizes and use colors. If one story is orange, then any other story that is similar in the quantity of work is also orange. When you encounter a story that is an appreciably different amount of effort then call it red. Repeat this until you have all your stories grouped. This process helps move one completely away from time and to start thinking about all work in relation to other work.

Now let's introduce numbers to the sizing process. It is important to note that this changes nothing about anything previously stated. The numbers are still an abstraction and almost strictly relevant to calculating velocity. One could easily translate the t-shirt sizes (xs, s, m, l, xl) to 1, 2, 3, 4, 5 and call it a day. The same could be said for the colors as long as they're ordered from smallest to largest. What many experienced teams will use instead of sequential integers is the Fibonacci sequence 1, 2, 3, 5, 8, 13, instead. These are still relative numbers. A 1 is less work than 2 while 3 is more work. The value here is in the meta-data. Teams tend to have a reasonably high precision when estimating a story as a 1, 2, or 3. What happens is that as stories get more complex our ability to accurately estimate diminishes. Something we might call a 4 could just as easily be a 5 or even a 6. For each step up the inaccuracy greatly increases. In fact, the general rule is that any story that is sized as an 8 is almost assuredly work that needs to become two or more stories and a 13 is a signal that we do not have sufficient information.

Velocity

Now that we are sizing our stories numerically, we can start tracking a metric known as velocity.

To properly explain velocity, there is another term that needs to be introduced, first. The Agile methodology says to pick a particular interval and to work on the most important stories during that interval. The interval is functionally no shorter than one work week and typically not longer than two weeks. Since we are iterating on the project work each interval, this period is sometimes called an iteration. In Scrum, which is a particular framework built on Agile principles, this interval or iteration is called a Sprint.

Each iteration the team and customers come together to identify the most important stories to be worked on. A story might be most important because it is integral to other stories. A story might be most important because it delivers urgent required functionality or other business value. The entire collection of stories that have been created are called the "Backlog". During the Sprint Planning selected stories are moved from the Backlog to the current sprint. This

Sizing

In the Waterfall (sizing) methodology one will define that they need to do some task and it will get added to the project plan. Next, the person will think about what they must do to complete that task and try to figure out how long it will take to do that work. This mental exercise quite frequently looks like this: *"I mostly know how to do 'x' and that will probably take me 1 hour. However, I must get some info from someone else, so I am going to throw a couple hours in there for that conversation and waiting for their answer. I also know that there is always some gotcha, so I am just going to take the 3 hours and double it. Between meetings and everything else, let's just call it one day."* Sound familiar? Sometimes the time estimate is right.

Sometimes the task takes way less time and other times it takes much longer. Despite this inaccuracy, we rarely change the time estimates. If it took longer, then everything dependent in the plan gets pushed out. If it took less time, then I get to do some of my other work to get caught up or try to get ahead.

Agile does not care about time at this stage. It is a fact that the work is going to take however long it takes and not a minute more nor a minute less. Instead, sizing in agile is an effort to quantify the amount of work that needs to be done. Much like the waterfall method, we look at a story and ask ourselves what needs to be done, but we do not attach a time to it. Instead, we are evaluating how much work is involved. Is it a small amount of work? Is it medium amount of work? Is it a large amount of work? Some agile teams actually size their stories based on t-shirt sizes such as XS, S, M, L, XL.

This abstraction of the amount of effort or quantity of work involved is frequently the most difficult part of sizing, both for people new to agile and even many that have been at it for a while. This is a good time to make an important note on agile sizing. Because this is not a measurement of time nobody should be asking themselves how long it would take if they were doing the work. Agile sizing assumes ideal conditions and that all knowledge, skills, and tools exist to do the work.

It would take me 10 hours or more to detail my vehicle. A professional could detail my vehicle in an hour. Notice that the actual work did not change, while the time changed by an order of magnitude. It is still "one detailing" whether I do it or the professional does it. What did change is the knowledge, skills, and tools.

At this point you might be wondering how much work is "medium" and who gets to decide. The answer is that medium is any amount of work that you and the rest of the team agree is medium. Going back to the vehicle detailing, if you and your team feel that Story 1 is medium then it is medium. If Story 2 is approximately the same amount of work as Story 1 then it is also medium. If Story 2 is less work, then it is a Small or eXtra Small depending on how much less work. Similarly, if it is more work than Story 1 then will be a Large or eXtra Large instead. There are a couple of important rules to keep in mind. First, the size does not change because you acquire the knowledge, skill, and/or tools that allow you to complete the task faster.

Second, effort in this usage is not about how long the work will take nor any lack of knowledge, skill, and/or tool. Even with perfect knowledge, skills, and tools some things just require more effort than others. Sizes are always based on ideal conditions.

We need to talk about one more concept in sizing before we move on to Velocity and finally answering the manager's question. So far, we have not used any numbers when talking about sizing and this is intentional. Agile sizing can be very uncomfortable until it becomes familiar. I would even encourage people new to sizing to step away from t-shirt sizes and use colors. If one story is orange, then any other story that is similar in the quantity of work is also orange. When you encounter a story that is an appreciably different amount of effort then call it red. Repeat this until you have all your stories grouped. This process helps move one completely away from time and to start thinking about all work in relation to other work.

Now let's introduce numbers to the sizing process. It is important to note that this changes nothing about anything previously stated. The numbers are still an abstraction and almost strictly relevant to calculating velocity. One could easily translate the t-shirt sizes (xs, s, m, l, xl) to 1, 2, 3, 4, 5 and call it a day. The same could be said for the colors as long as they're ordered from smallest to largest. What many experienced teams will use instead of sequential integers is the Fibonacci sequence 1, 2, 3, 5, 8, 13, instead. These are still relative numbers. A 1 is less work than 2 while 3 is more work. The value here is in the meta-data. Teams tend to have a reasonably high precision when estimating a story as a 1, 2, or 3. What happens is that as stories get more complex our ability to accurately estimate diminishes. Something we might call a 4 could just as easily be a 5 or even a 6. For each step up the inaccuracy greatly increases. In fact, the general rule is that any story that is sized as an 8 is almost assuredly work that needs to become two or more stories and a 13 is a signal that we do not have sufficient information.

Velocity

Now that we are sizing our stories numerically, we can start tracking a metric known as velocity.

To properly explain velocity, there is another term that needs to be introduced, first. The Agile methodology says to pick a particular interval and to work on the most important stories during that interval. The interval is functionally no shorter than one work week and typically not longer than two weeks. Since we are iterating on the project work each interval, this period is sometimes called an iteration. In Scrum, which is a particular framework built on Agile principles, this interval or iteration is called a Sprint.

Each iteration the team and customers come together to identify the most important stories to be worked on. A story might be most important because it is integral to other stories. A story might be most important because it delivers urgent required functionality or other business value. The entire collection of stories that have been created are called the "Backlog". During the Sprint Planning selected stories are moved from the Backlog to the current sprint. This