

Mind Mapping for Agile Learning



Daniel W. Rasmus

Principal Analyst
Serious Insights LLC

Contents

Introduction:	
Mind Mapping for Agile Learning	3
Agile Learning:	
Why Mind Mapping?	4
Agile Learning:	
Agile Techniques	5
Agile Learning:	
Learning with an Agile Mindset and Scrum Toolset	6
A Worked Example of Agile Learning:	
Romeo and Juliet	8
Scrum for Teams:	
Understanding Roles	11
Scrum for Teams:	
A Note on Learners as Customers	12
Agile Learning:	
Next Steps	13
Sidebar:	
From Lesson Plan to Agile Learning Sprint	14

Mind Mapping for Agile Learning

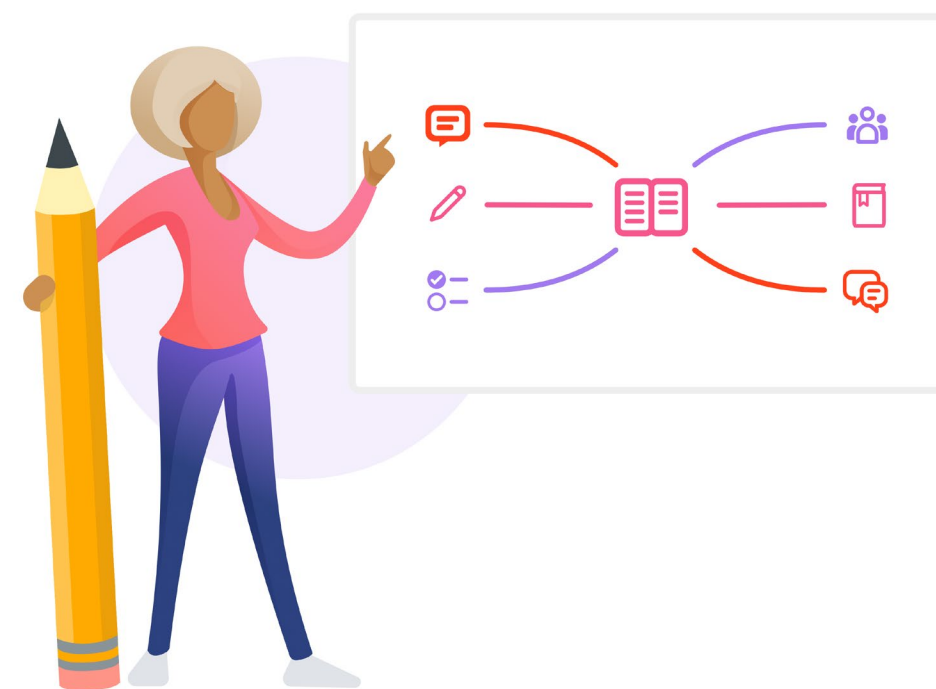
Mind mapping has always been my go-to tool for lesson planning. In fact, I have never created a lesson plan that didn't start with a mind map. Unlike linear codifications of learning that become static when printed or saved to a PDF, a mind map retains an openness to new data, new ideas, and new knowledge. They are living, evolving learning aids.

This flexibility has proved valuable. Since the onset of the COVID-19 pandemic, educators have reinvented curricula for remote learning, quickly mastering new technologies to smooth the transition to remote learning models. Modern collaborative mind mapping, powered by tools like MindMeister, has facilitated the rapid reshaping existing material. More importantly, however, I believe mind mapping to be central to a more dynamic teaching approach that will be valid even after the crisis ends: Agile Learning.

This theory stems from my background advising technology companies and IT organizations, predominantly in a discipline known as Agile Development. Agile is a set of principles that facilitates faster, customer-oriented development. Business analysts, developers, product owners, and others use short iterations to make improvements to a product. They learn as they go, adjust plans as new insights arise, and incrementally move toward delivery. Mind maps often act as the starting point for Agile planning, with task management tools taking over the bulk of the workload as projects move from concept to execution.

I realized that I could reimagine many teaching and learning experiences within the Agile framework. Static lesson plans could be made dynamic, with learners empowered to co-create learning experiences with learning goals and products to develop instead of notes, worksheets, and projects. Agile Learning offers a more connected learning experience that shifts responsibility for discovery to the learner, inspired by ideas like the flipped classroom.

This paper outlines how mind maps can facilitate the transition to Agile Learning. As an example, I adapted publicly available lesson plans on Shakespeare's Romeo and Juliet to the approach. The results are high levels of learner engagement, new interaction models for educators, a learning journey augmented with practical skills like project management, collaboration, and facilitation. These, I hope, will prove beneficial to learners in their lives beyond school.



Agile Learning: Why Mind Mapping?

Agile Learning requires dynamic tools. It does not rely on a predetermined list of tasks, rather on a set of goals, objectives, and products; empowering learners to discover methods and approaches that will fulfill those objectives. Mind mapping easily captures the dynamism of brainstorming by embodying the execution and iteration at the core of Agile Learning. Mind maps easily contract and expand as necessary for exploration, offering focus and context to users.

Modern mind mapping tools, like MindMeister, are collaborative. While pen-and-paper mapping creates a contemplative, intimate, and personal experience, collaborative mind mapping compels people to work together.

The latter point is especially relevant to Agile Learning, which requires flexible representation. Although teams never know what they may encounter in their learning journey, tools like MindMeister allow them to collate research links, images, quotations, sketches, notes, and even tasks in one place.

While linear lesson plans require repetition to associate ideas, mind maps are a more natural convergence of connected content.

This project planning capabilities of mind mapping tools inspired companies like Meister, the creators of MindMeister to expand beyond mind mapping into task management with their online tool MeisterTask. The phrase “from thought to action” describes a process that dovetails with a principle at the heart of Agile Learning: a seamless transition from of ideas from the mind map to structured project boards that guide research and learning development. The mind map becomes a home for ideas, a launch pad for the implementation of learning goals.

Complex structures like lesson plans introduce interconnected topics. While linear lesson plans require repetition to associate ideas, mind maps are a more natural convergence of connected content. Multiple ideas may refer to a single-entry point, making them the ideal canvas for ensuring that standards get met and goals achieved.

The fluidity of mind maps often overshadows another valuable attribute: compactness. Digital mind maps can be sprawling, multi-page artifices to creative thinking, planning, or research—but they can easily collapse back to a single set of high-level concepts. Each topic can be expanded or hidden when needed, on any device, anytime. This enables mind maps to support the needs learners by adapting to their needs.

A compact mind map facilitates underlying content organization. For example, with linear methods, references to love and marriage in *Romeo and Juliet* would be split across several lessons. However, the re-articulation of the lesson plan as a mind map allows the reorganization of complementary ideas in a more visual style. The mind map facilitates the rapid integration of connected ideas by allowing users to drag-and-drop them to new positions, combine them into a single lesson, or perhaps prune them to save time.

Digital mind maps can aid ideation too, either projected onto a wall during a team meeting or lesson, or as a joint brainstorming tool when an event sparks a train of thought.

The technology has transformed how many people work and learn. In education, lesson planning, teaching, engaging students, and tracking complex information about practices and standards are all valid use cases.

For this paper, mind maps are depicted as the starting point for agile methodology in the classroom. They help learners self-manage collaborative learning journeys, stimulated and driven by ideas that that reflect the unique nature of each student.

Agile Learning: Agile Techniques

It is important to understand a few of the concepts related to agile software development, and why they are so applicable to the learning experience.

The delivery of software in most modern businesses, including commercial app developers, draws on a technique called Agile Development. At its core, Agile Development focuses on the delivery of a product with a defined set of features. It relies on teams of individuals to deliver an increment of software that offers value to a customer. Large software systems consist of hundreds or thousands of increments.

Figure 1 simplifies the Scrum flow to a single workflow. Customers and developers cycle through the same steps, over and over, adding value through features implemented during each iteration. The work to complete an iteration is called a sprint.

In Agile Learning, sprints become individual learning increments, each focused on some aspect of the subject. In our *Romeo and Juliet* example, this might be the play's themes, the history of Elizabethan theater, or character attributes.

Agile Work happens through a framework called Scrum, a term borrowed from rugby that encourages experiential learning and self-organization focused on problem-solving and continuous improvement.

The development of software starts with a product owner. This individual represents a community of users with a desire to develop a system with a specific function, such as find restaurants or analyze student grades. The product owner shares a prioritized list of ideas with a Scrum team. The ideas are broken down into a series of sprints that offer compact, relatively easy-to-achieve divisions of work.

In Agile Learning, the teacher becomes the product owner, driving expectations for learning achievement. However, Agile Learning empowers the learner to navigate the learning, figure out how best to represent their knowledge, and meet the learning goals.

Agile and Scrum complement each other. Agile Development embodies a set of principles (found [here](#)), while Scrum offers a technique for getting work done. They combine well because Agile suggests the need for rapid iterations and delivery of work, and Scrum offers a good way to achieve that goal.

They also fit well in classrooms that seek to engage, empower, and co-create learning experiences.

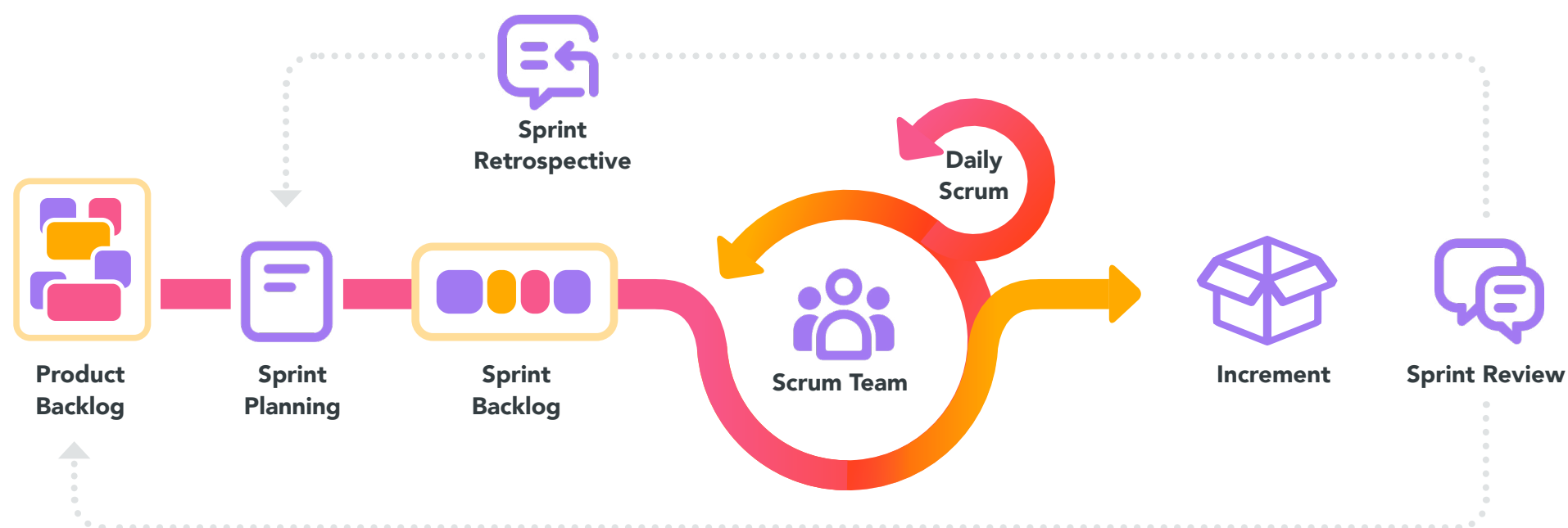


Figure 1:
Scrum Framework
(after Scrum.org)

Agile Learning: Learning with an Agile Mindset and Scrum Toolset

At a conceptual level, learning starts with a set of goals for learners. Lessons should help the learners achieve that level of competency.

That same basic premise holds for software development. Here, the goal is to deliver an increment of software to a project or a product at a level of competency and completeness, with an associated perceived value.

Agile development differs from traditional software delivery in that it does not begin with an overarching plan. Agile projects start with a general idea in which many aspects remain unknown to the product owner and development team. In other words, Agile is a learning experience.

Like traditional software development, traditional learning starts with a plan. For most educators, the plan will be captured in a lesson plan or syllabus that outlines the entire learning journey, how grades are calculated, and other aspects of assessment and outcome. For the student, however, even if provided with a full course description, the course begins for them as a mystery. In some courses, even the language of the overview may challenge their current knowledge.

Agile learning's incremental approach aligns well with the presumption of the unknown. The assignment is not to execute on a set of tasks and build a piece of software, but to co-evolve an understanding that leads to a product which meets customer needs. By doing so, it offers value to those customers.

Learners can benefit from the Agile approach. I teach a course on social media for jobseekers, which is primarily an experiential lab. There is no right or wrong answer to most questions, for instance how to craft a LinkedIn profile or write a tweet.

It does, however, require a certain degree of technical know-how, such as where edit fields lie and what preferences to set, but those also change regularly—as does the learner's work experience and their target market for employment.

Most learners come into the class with little social media experience. I attempt to create a co-learning environment that gives them an opportunity to use real software, engage platforms and their users, and gain knowledge about both.

The learners explore what is unknown to them, gaining insight and proficiency as they explore. They write their profiles at least three times, each one incrementally better, richer, and more detailed, as they learn about how to craft their story as well as how to connect that story to their ever-evolving work skills.

While the class is not formally a Scrum, it leverages several techniques that break down work into smaller chunks. These bring closure, learning, feedback, and purposefully set up the next phase of learning. Explorations of personal values inform an understanding of how to create a personal brand, which then becomes the framework for representing oneself on the Internet.



As you will see in the worked example, typical daily lessons can be broken down into sprints. While it is not common to pre-think sprints in a development environment, this approach ties traditional lesson planning to the incremental approach. Teachers should be open to modifications of their plans based on reflection and feedback from learners.

Scrum also includes metacognitive steps, called retrospectives, that offer improvements to process or product. New features or ideas may emerge during execution that end up getting put on the sprint backlog.

Opportunities for process improvement can be implemented as quickly as the next sprint, accelerating continuous learning into continuous improvement.

Each day, learners gather around their sprint backlog to see what needs to be done to complete the sprint.

In Agile Learning, learners gather around their sprint backlog each day to see what they need to do to complete their sprint. Lessons may be executed more quickly or take longer than planned, depending on sprint progress. Quality, rather than time, is the primary driver in a sprint, though the selection of work offers some commitment to schedule. Quality derives from what the product owner, in this case, the teacher, states as a view of what “done” looks like. In the worked example, you will see that rubrics have been translated into descriptions of “doneness”.

In a traditional learning environment, grades are the ultimate and lasting expression of quality and completeness. Rubrics exist to offer learners insights into their performance. In contrast, rather than a gradation of quality of understanding and adeptness, the worked example adopts the rapid failure, learning, and continuous improvement values from Agile.

An increment continues to be worked on by the team until it meets the “done-ness” criteria. I selected the rubric associated with an “A” as it represents the best version of doneness.

Boards, or task lists, capture the set of actions associated with delivering a learning sprint. These boards are adopted from the Japanese word “Kanban” which means “visual board.” In Agile Learning, these boards become the task list, the place where everybody goes to see that status of a sprint, what is happening now, and what remains to be done. You can read more about boards and task management in the sidebar: [Exploring Sprints](#).

The diagrammatic representation of Scrum oversimplifies the process. Sprint planning may be complex. Learners, for instance, faced with representing a historical timeline, are not simply asked to fill in a pre-printed timeline but rather to also solve for how to represent events. They not only gather knowledge but must communicate what they learned, including gathering the material and creating the representation. “Doneness” goes beyond recall or completeness to include an understanding of how the knowledge impacts the learner, the next steps in the projects, and how their potential customers might use it.

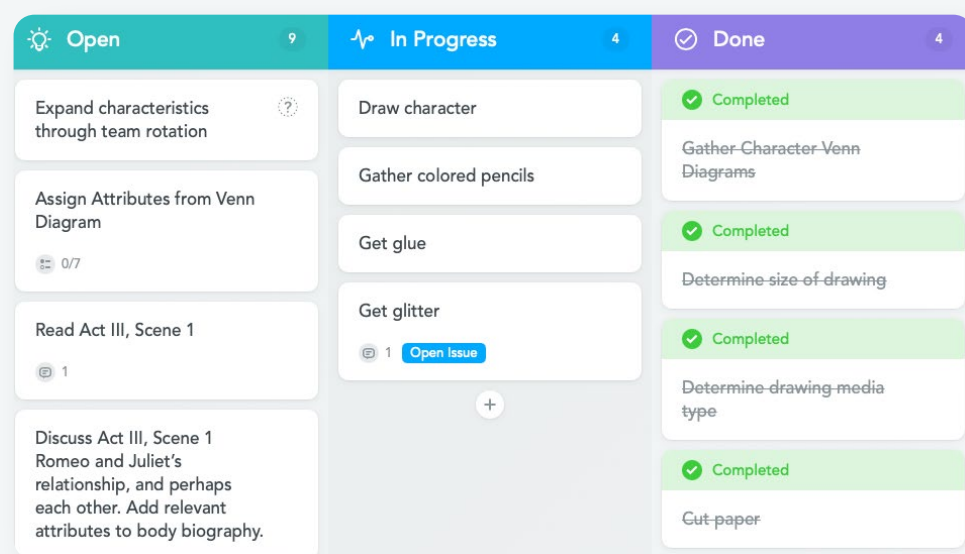
In a nod to bringing future work skills into the learning mix, the worked example does not end with a student presentation of a final product but with the creation of “marketing materials” aimed at selling the “product.” These marketing materials help the student reflect on the totality of the learning experience and its value and express that value to a hypothetical audience of customers who might buy their creations.

In classrooms that adopt this approach, these marketing materials might well be used as input by the product owner, the teacher, for the next learning journey through the material. Not only does continuous improvement apply to the product and processes during a lesson, but between lessons, year over year.

The worked example will further explore key Agile and Scrum ideas as its narrative unfolds.

Exploring Sprints

A sprint captures all activities a team will spend its time on, including research, material gathering, purchasing, drawing, coloring, ordering, and other actions. The learning backlog acts as the foundation, but the sprint expands this basic idea into steps to deliver on the requirement. This engages learners in activities that, while not directly applicable to the lesson, offer additional skills. Adopting Agile and Scrum approaches allow educators to expand single discipline subjects as with literature in this example, into other areas like leadership, organization, planning, etc. This level of detail also expresses how seriously learners are taking a sprint. In engineering, the level of detail helps during the negotiation for additional resources, as the detail backlog demonstrates actual work as opposed to conceptual work. The level of detail is critical to the daily Scrum as the team looks to prioritize work, identify obstacles, and reorder the work based on available materials or other resources. The educator can also visually see progress and engagement. These boards are also the key interaction point from the acting Scrum Master to help teams better manage, as well as work through bottlenecks or barriers to success. The figure below offers a view of a single sprint's board.



A Worked Example of Agile Learning: *Romeo and Juliet*

For the worked example, I converted a publicly available lesson plan for Shakespeare's *Romeo and Juliet* into a mind map that models a series of sprints through the material. (See the *Converting a Lesson to A Sprint* sidebar). I chose *Romeo and Juliet* first based on familiarity. Most people reading this paper have a basic knowledge of the play and likely will have studied it at some time. I study Shakespeare extensively and wanted to build an example that would challenge me to think about the play in a different way. I chose a comprehensive, ambitious, six-week-long lesson plan as the jumping off point. The lesson plan's rich exploration of the material provides abundant entry points for reimagining the traditional techniques as team-based, co-creation experiences (My thanks to Lynn Berry, Sarah Donovan, Meghann Hummer for their ELAN 4400 Conceptual Unit dated December 2, 2003).

Figure 2 offers a revised view of the Scrum process as applied to learning. This diagram will act as the basis for the worked example.

In addition to the learning activities, each lesson listed associated standards. During the process of reimagining the lessons for Agile Learning, these standards were transformed into what the agile development community calls *user stories*. User stories act as the voice of the customer—a set of expectations.

Most software features you find on your smartphone start their journey as a user story that specifies some experience or outcome desired by a product owner on behalf of a customer. "Customers should be able to read a PDF on their phone" is an example of a user story.

User stories illustrate intent for those charged with delivering value. User stories, rather than the arcane language of standards, can help ground learners in expectations meaningful to them. The user stories also acted as a scaffold for exploring how to capture relationships in the mind maps.

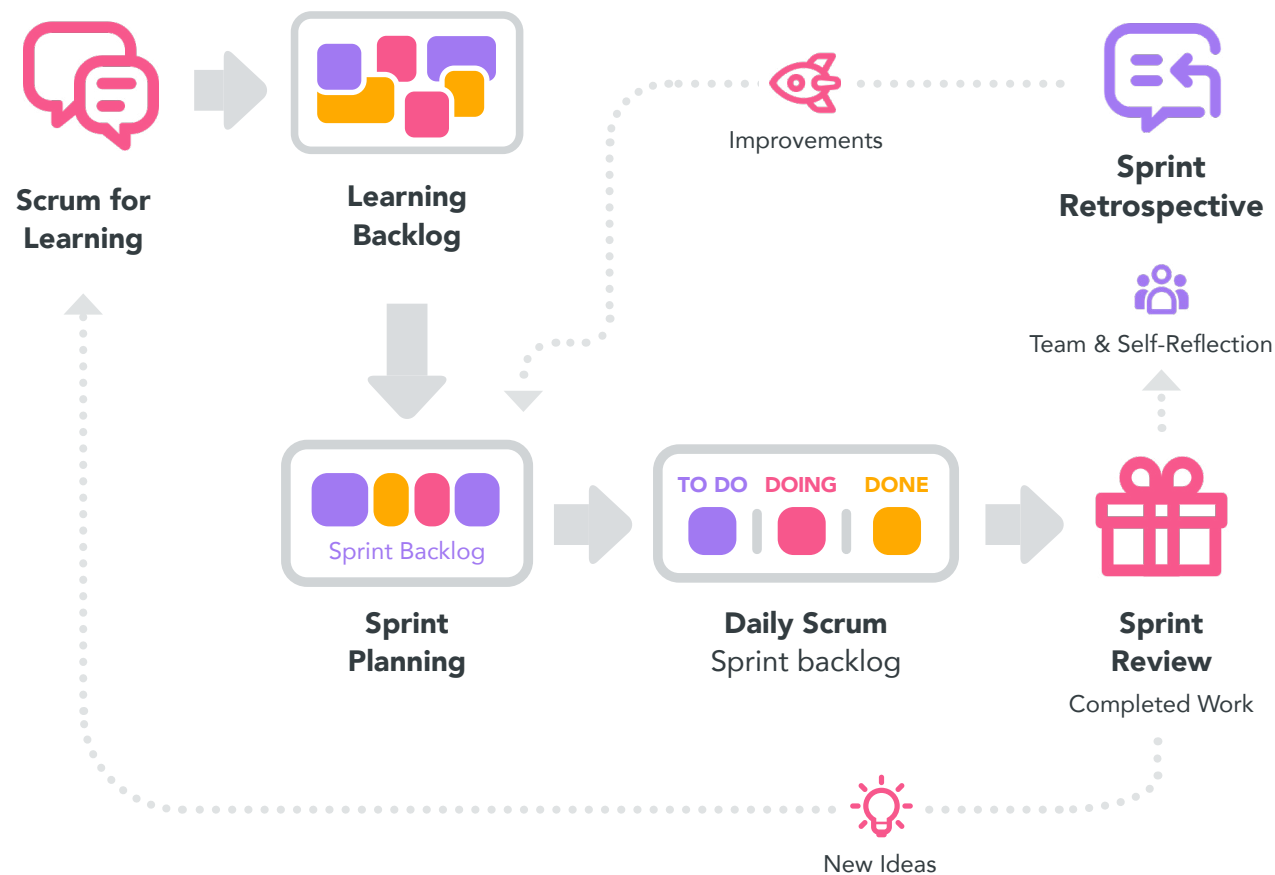


Figure 2:
A revised diagram of Scrum applied to learning. (Source: Serious Insights LLC).

Vision: The extensive essay at the beginning of the sample lesson plan set out context and goals in narrative form. I condensed the narrative into a vision for the learning experience. The vision informs the goals and the day-to-day learning experience, as well as the products of that experience.

Learning backlog: The product backlog captures the features of the product. These have been primarily listed as outputs. User stories have been captured separately. However, a user story could easily expand to include the associated sprint, for instance: *Students should expand their vocabularies through reading, the study of etymology, the use of dictionaries, and other references to the types of love in Romeo and Juliet.*

The learning backlog order has been left a bit raw to allow those applying the mind map to select among features more easily without feeling that the organization of the outputs overtly prioritizes their inclusion.

Sprint planning: Sprint planning typically begins by slicing off a consumable portion of the product backlog. For example, creating a Shakespeare Dictionary, a set of unique Shakespearean words and their definitions could be selected in one iteration.

Sprints: The mind map offers several sprint examples. Sprints are driven by the learning backlog. Learners take on a piece of work, such as creating a representation of Shakespeare's biography, as an increment of work. Over time, they work through all the items in the learning backlog. Each of these becomes a sprint. As they execute the work, Sprint Retrospectives may suggest improvements that add new items to the list. Co-created learning implies that learners are responsible for their learning and for providing feedback. They may well discover new facts that require changes to the assumptions underlying the learning intent.

The sprints in the accompanying mind map should be considered prototypical. This is because their order, while dictated by the order of the original plan's lessons need not fall into this sequence during execution. Plays lend themselves to linear exploration, but the work on the sonnet, for instance, need not take place where sonnets arrive in the play. It may make more sense to include the sonnet as a deep exploration of a literary term, along with the introduction of auxiliary material from Shakespeare and his contemporaries. Likewise, the details in the sprints offer prototypical activities. Learners should be encouraged to approach the requirements creatively, perhaps with little guidance at the onset. They may create new approaches to the material that early constraints or encouragement toward an approach may curtail.

Sprints may also be combined or take place more quickly than a lesson plan may forecast. Some short sprints may also reach their objectives in less time than anticipated.

Sprints can be visualized on task boards as the sprint backlog. A tool like MeisterTask, also provided by Meister, represents these tasks electronically. Before the advent of electronic project boards, whiteboards or walls were peppered with sticky notes to capture the tasks, their order, and their state.

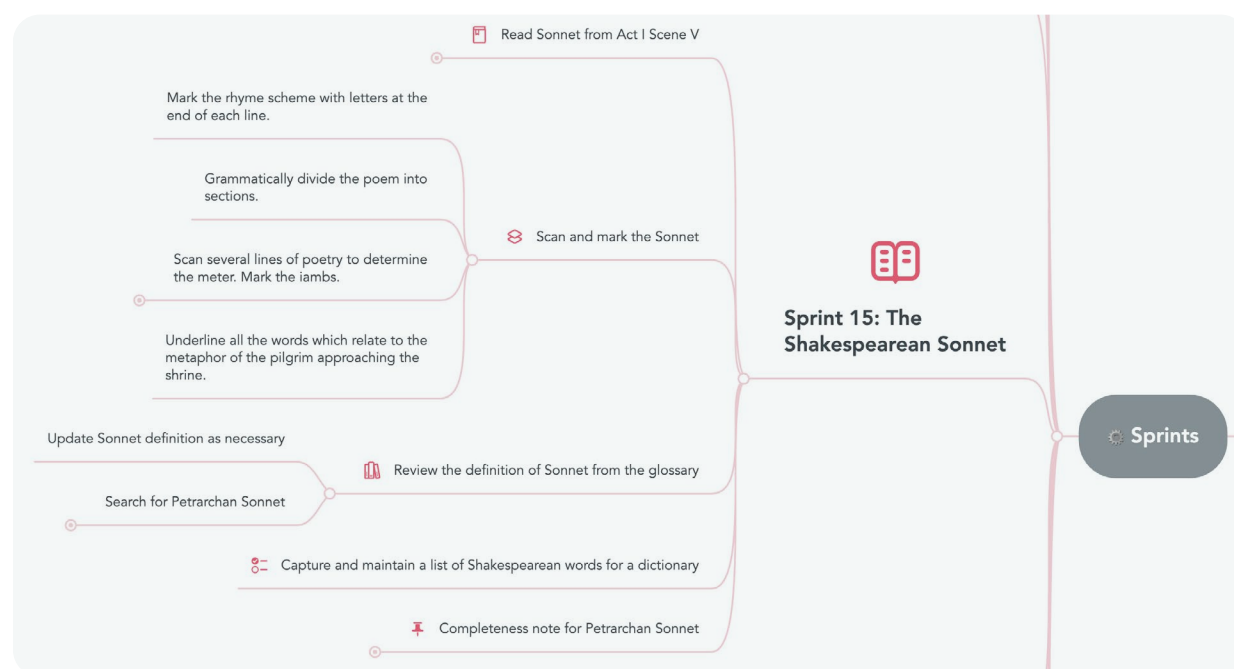


Figure 3:
A mind map example of a sprint for understanding sonnet structure.

Sprint review: Each sprint ends with a **sprint review**, which typically demonstrates the product. For example, the “Compare and Contrast Two Types of Love” essay would be presented, along with how it used the previous sprint as a resource. The product owner would review the essays for completeness and value. Do these *demonstrate how to compare and contrast objects, groups, or concepts in writing?* If the answer is no, the sprint may have failed, but the unfulfilled requirement would simply return to the list to be tackled again. The iterative approach means that students, as developers, can make multiple passes at achieving a valuable output.

The sprint review is not a task but a conclusion to a task. It is not in the sprint backlog; it simply happens at the end of a sprint as the team reviews its work with the teacher.

Each sprint also requires a reflective activity that includes a consideration for the team’s performance and sprint outcome. The reflection may generate new ideas that end up on the sprint backlog. The existing requirements may return to the board with additional clarification so they may be more adequately addressed in a future sprint. Team members should see the reflection exercise as an opportunity for meta-cognition. They should think about what they learned and how they learned it.

The central philosophy of agile comes down to creating valuable experiences through rapid, iterative work that often involves failure and learning.

This lesson plan’s six-week unit was selected for its scope. Educators need not be as ambitious in their own plans. Deselecting certain features, such as the Literary Terms Game, or the Elizabethan Concept Map, immediately simplifies the product and reduces the number of sprints. On the other hand, there are other ideas in the play that are not captured in this lesson, such as the chemistry of poison, which may be helpful for an educator seeking a more integrated lesson beyond literature, reading, and history. The play’s prologue refers to the young lovers as “star-crossed,” which offers an opportunity to discuss astrology and how non-scientific beliefs continue to influence the human experience.¹

This worked example would surely change if implemented in a classroom. And it should. The central philosophy of agile comes down to creating valuable experiences through rapid, iterative work that often involves failure and learning. It differs from traditional lesson planning by not trying to capture every nuance and possibility at the onset. Instead, it employs a core set of ideas and expectations to drive innovative ways to meet those expectations.

¹ Also consider Mercutio, who may well be named for the planet Mercury which is the constellation Gemini’s planetary rule. Gemini is a zodiac sign that contains a double, or pair, that may result in coherence or discord.

Understanding Roles



Product Owner

The key stakeholder who represents learning objectives and standards. In Agile Learning, the product owner is typically the teacher.



Scrum Team

A team that completes the committed-to work within a sprint.



Scrum Master

The Master ensures the team is productive by helping the team with processes, removing impediments, protecting time, and eliminating distractions.

The role of the Scrum Master requires further elaboration. In Scrum practices, the Scrum Master comes to the process with a leadership role. The Scrum Master is typically a certified expert in Scrum, with strong management and leadership skills. Some practitioners recommend that a Scrum Master hold the role for the entirety of a project. For many Scrum Masters, facilitating Scrum Teams becomes their only job.

In a learning environment where the learners are the Scrum Team, schools will unlikely certify a Scrum Master to focus on students learning Shakespeare. Some may employ a Scrum Master, but they will be spread thin, acting more as a general coach than an individual involved in active learning. Someone,

however, needs to own the Scrum Master role to help the team along, and that can't be the teacher, as the teacher is already occupied as the product owner, the voice of the customer.

Teachers should establish a basic grounding in Scrum principles by offering a Scrum Master introduction ahead of applying Scrum to learning. Schools adopting this idea should offer support and training to students and faculty.

For the Scrums, a student should volunteer or be assigned as a Scrum Master for each sprint, rotating on an agreed-upon frequency.

In business, Scrum Master is a paid role with professional expectations assigned to it. However, in a learning situation, acting as a Scrum Master, even if occasionally in inexpertly, offers an enormous learning opportunity that would not otherwise present itself.

A Note on Learners as Customers

The point about Scrum Masters learning on the job is important. Unlike commercial uses of Scrum, with external customers or internal business functions, the student is the main customer in Agile Learning. Students perform in the Scrum team and benefit from its outcomes. Their work brings them new subject knowledge and enhances their collaboration, feedback, meta-cognition, and work process management skills.

Most learners today do not get the opportunity to choose how they learn. Instead, a set of instructions programmatically sets out a learning path that they follow. How well they follow that path influences their success or failure in learning assessments.

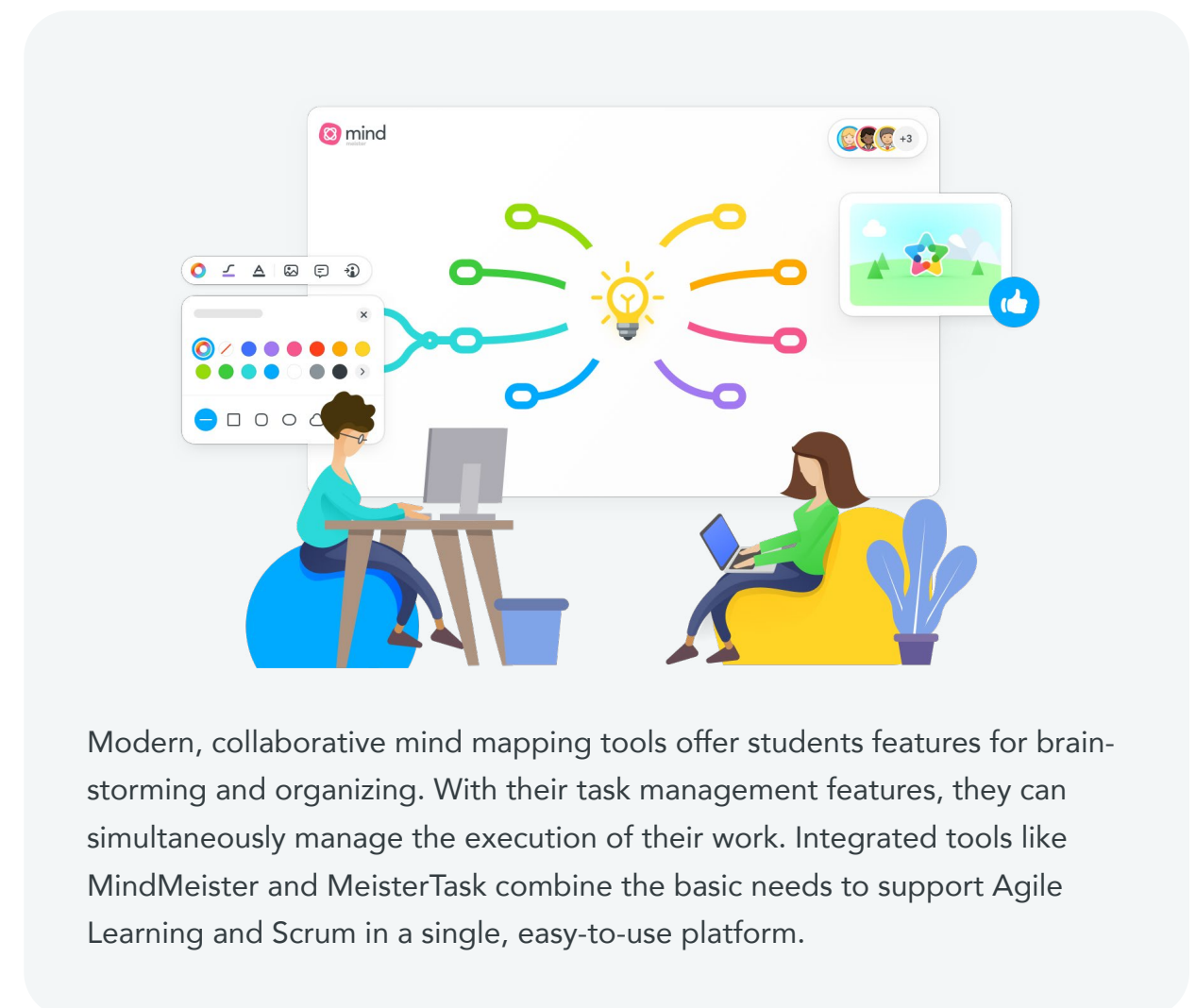
In contrast, by adopting Agile Learning, learners co-create the learning experience. They may choose how to approach a task, not just as an individual, but as a member of a team. They learn how to build consensus, advocate for their ideas, and how to recognize that another idea may be more robust than their own. They learn how to combine two or more ideas into an even more effective one. Individual, personal learning does not force learners to confront these topics in the same way.

Early in this paper, I suggested that students enter new subjects with a sense of mystery. Unfamiliar topics creates stress and introduce uncertainty about control over learning. By creating incremental learning opportunities, students can explore ideas without becoming overwhelmed. If incremental improvements and continuous learning accompany the experience, students quickly discover that occasional failures are simply part of the path toward good outcomes.

The standards stated as user stories apply to the learner. Each sprint reflects an approach to teaching, an aspect of writing, or literary analysis. This approach elevates and augments learning experiences by asking students to consider how the product of their work might be used as potential learning tools for others.

It should be stated that the teacher is also a customer of the learning process. With extensive and regular feedback, teachers gain more insight into how learning works—and how well their rubrics apply in a transparent, open, and co-created learning environment.

The first principle in the Agile Manifesto states, “Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.” To apply that principle to learning, we simply change *software* to *learning experiences*. Transforming standards into user stories, which learners adopt as drivers for learning, creates holistic experiences. The tensions between expectations and performance relax when everybody tries to achieve the same goals.



Modern, collaborative mind mapping tools offer students features for brainstorming and organizing. With their task management features, they can simultaneously manage the execution of their work. Integrated tools like MindMeister and MeisterTask combine the basic needs to support Agile Learning and Scrum in a single, easy-to-use platform.

Agile Learning: Next Steps

As an analyst, I worked with several organizations that faced extreme uncertainty during the COVID-19 pandemic. Many required new operating models and approaches to work. Educators across the world experienced the same extremes and disruptions as commercial businesses. As a result, education too looked for ways to employ continuous learning and collaborative, adaptive responses.

Agile Learning offers an approach to engaging students, working in teams from home, or exploring subjects together in the classroom. Similarly, the use of mind mapping for Agile Learning means that the process is more visual, collaborative, and easier to adapt: a natural extension of the principles of Agile.

Start by experimenting. Find a subject that would benefit from a fresh teaching approach and try Agile Learning. Every implementation will be different, but the technique will invigorate the learning experiences in practically any subject area.

Take a lesson plan and mind map it. Then, start thinking about how to reshape it into goals and products. Most importantly, consider how to transform activities into opportunities for discovery, and lectures into completeness criteria.

Our dynamic world demands approaches like Agile Learning.

Throughout the process, be open to learning. There is no subject that doesn't continue to expand its horizons. Physicists and cosmologists still struggle with the foundations of physics, new artifacts regularly update our understanding of history, and new inventions change our relationships with ideas, the world, and each other.

Our dynamic world demands approaches like Agile Learning. The tools, like collaborative mind mapping, exist to support the transition from traditional lectures and homework into the creation of dynamic, distributed teams of learners who co-create their learning experiences.



Sidebar:

From Lesson Plan to Agile Learning Sprint

User Stories

Step 1. In the center of Figure 4, between the two approaches, the standards translated into user stories are listed. Note that the first standard, “identify protagonists and antagonists and their motivation” does not appear as an associated user story for this lesson, as the sprint does not support this. That user story may exist, but this sprint will not help achieve it.

Teachers should consider translating standards into user stories before creating their learning backlog. The two become a bi-directional check on each other. A learning backlog should support all user stories. If it does not, it means that a learning outcome has been missed. It may also mean that this product/project does not support that user story. Delete that user story/standard and address it in another product.

The Traditional Lesson Plan

The left of Figure 4 captures a traditional lesson plan. It includes a learning concept, goals, materials, and procedures. Because the teacher presumes involvement and direction, it specifies how the student will represent attributes associated with characters, a Venn Diagram, and suggests the teacher will explain Venn Diagrams.

The lesson also includes content about Romeo’s love for Juliet with a journal assignment focused on taking a position and explaining any skepticism the learner may feel. This has been eliminated from the sprint as it does not directly inform the character attributes or help produce the representational product.

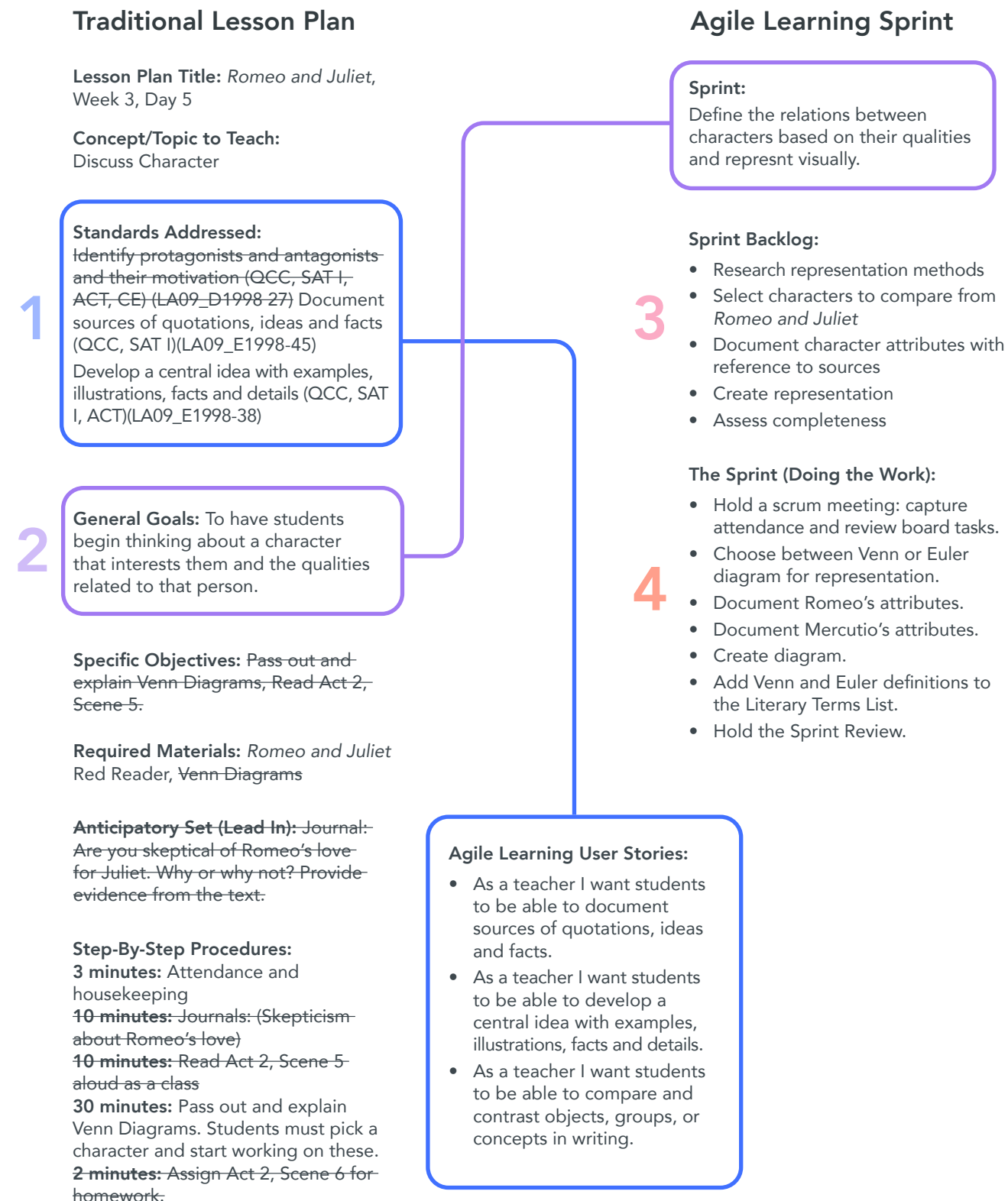


Figure 4: Information adapted from Lynn Berry, Sarah Donovan, Meghann Hummer ELAN 440 Conceptual Unit, December 2, 2003

The Sprint

The right side of Figure 5 offers a potential translation of the lesson plan into a sprint.

Because Scrum teams are self-managing, the housekeeping of who is in attendance and what they need to do next becomes the team's responsibility. Because the Sprint board is visible to the teacher, the state of sprints across the teams is transparent.

Step 2: Define the sprint. The sprint action to "Define the similarities and differences between characters based on their qualities and represent them visually" Replaces the vague topic of "discuss character." It became clear from the onset that the goal is some type of illustration.

Step 3: Break down the work. The sprint backlog lists activities required to meet the sprint's stated goal of a visual representation of character similarities and differences. The single step-by-step procedure to "Pass out and explain Venn Diagrams. Students must pick a character and start working on these" from the original lesson plan becomes an explicit set of activities. Namely, to select characters to compare, to document their attributes with references to the sources, and to create the diagram. The goal is not to start working on them, but to finish them. Each sprint results in completed work as the demonstrable value, in this case, the results of research and analysis on two of Romeo and Juliet's characters.

The sprint on Venn or Euler diagrams reflects a co-creation example. It is easy to hand out a Venn diagram, explain the idea, and then ask students to fill one out for two characters. It could be argued that it would be more engaging to ask students to figure out a way to visually represent character attributes. In their research, they would discover Venn and Euler diagrams. Either would serve as an acceptable representation for this exercise. This would suggest that the user stories may be expanded to include one on "compare and contrast", as the students would need to justify, at least to each other, which approach would best meet the task requirements.

Step 4: Do the work. "The Sprint (Doing the Work)" would not exist on a board in an Agile Learning environment. However, I included it to reference the actual work done during the sprint. At the end of the sprint, the team would offer a Euler or Venn diagram for review, with character names, attributes, and references. During the sprint review, the teacher may compare a set of character attributes against the student's diagrams and discuss any significant discrepancies. The teacher might also note meaningful insights that add to their own knowledge base to be used for future classes.

Note that the last item on the list refers to a previously completed "Literary Term List" sprint. Because the students have ongoing improvement tasks for previous "features", they may incrementally add new content to previous work. In this example, the sprint resulted in research on Venn and Euler diagrams, which informs them of which to use for the visual representation. Rather than lose the work, it becomes an entry on the Literary Terms List.

Figure 5 illustrates a view of the Sprint in process within a MindMeister mind map.

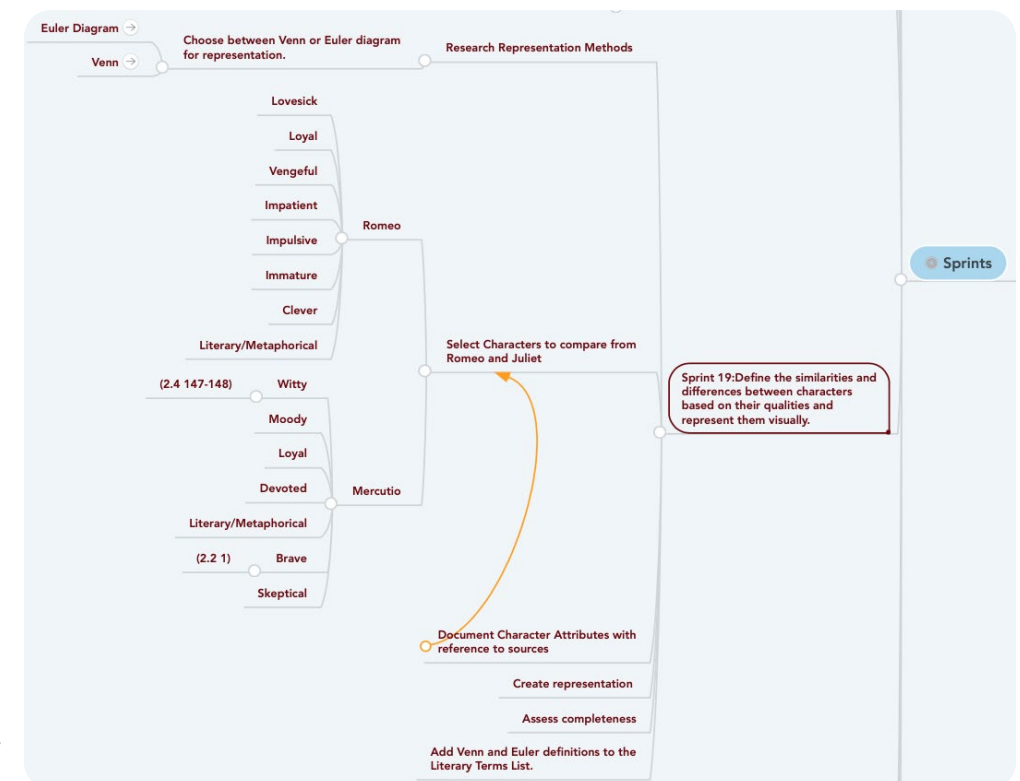


Figure 5:
the Sprint in process
within a MindMeister
mind map.



About the Author

Daniel W. Rasmus is the Founder and Principal Analyst at Serious Insights LLC. He is the author of *Listening to the Future* and *Management by Design*. His writing on learning, management, and the future of work has appeared at *HBR*, *FastCompany*, *University Business*, *Wired*, and other publications.



Serious Insights

Serious Insights is a boutique analyst firm located in Seattle, WA, that serves vendor and end-user clients through coaching, content development, and scenario planning-based workshops on the future of work and learning.

For more information visit:
www.seriousinsights.net



Meister

Meister creates stunning online productivity and creativity solutions that people simply love to use. Each tool in the Meister Suite integrates smoothly with the others to ensure flawless workflows and keeps teams working together, wherever they are, whatever they do.

For more information visit:
www.meisterlabs.com