



**CAPSTONE**  
Expect Solutions • Expert Resources

# Capstone Agile Model (CAM)



---

## **Capstone Agile Model (CAM) Approach**

Everything we do within the Capstone Agile Model promotes a disciplined project leadership process that encourages frequent inspection and adaptation, a leadership philosophy that encourages teamwork, self-organization and accountability, a set of engineering practices intended to allow for rapid delivery of high-quality software, automated testing framework practices, and a business approach that aligns development with customer needs and company goals. .

### **Definitions**

CAM consists of three components: Project Leadership, Engineering, and Quality Assurance.

### **Project Leadership**

Our Project Leadership methodology is rooted in Scrum and builds upon industry leading software engineering practices.

Scrum is an iterative, incremental framework for project management and agile software development. Scrum was first conceptualized in 1986 by Hirotaka Takeuchi and Ikujiro Nonaka, to describe a new holistic approach that would increase speed and flexibility in commercial new product development. They compared this new holistic approach, in which the phases strongly overlap and the whole process is performed by one cross-functional team across the different phases, to rugby, where the whole team "tries to go to the distance as a unit, passing the ball back and forth." The case studies came from the automotive, photo machine, computer and printer industries.

In 1991, DeGrace and Stahl, in "Wicked Problems, Righteous Solutions," referred to this approach as Scrum. In the early 1990s, Ken Schwaber used an approach that led to Scrum at his company, Advanced Development Methods. At the same time, Jeff Sutherland, John Scumniotales, and Jeff McKenna developed a similar approach at Easel Corporation and were the first to call it Scrum. A few years later, Sutherland and Schwaber jointly presented a paper describing Scrum at OOPSLA '95 in Austin, TX, its first public appearance. Schwaber and Sutherland collaborated during the following years to merge the above writings, their experiences, and industry best practices into what is now known as Scrum. In 2001, Schwaber teamed up with Mike Beedle to describe the method in the book "Agile Software Development with Scrum."

### **Engineering**

Our Engineering methods draw on multiple practices including creating domain models and class diagrams, code reviews, and continuous integration/regular automated builds. These engineering methods ensure that lightweight and high quality code is written by the developers.

Code review is systematic examination (often as peer review) of computer source code intended to find and fix mistakes overlooked in the initial development phase, improving both the overall quality of software and the developers' skills.

### **Quality Assurance**

Continuous Integration (CI) implements repeated processes of quality control - small pieces of effort, applied frequently. Continuous integration aims to improve the quality of software, and to reduce the time taken to deliver it, by replacing the traditional practice of applying quality control after completing all development. CI involves integrating early and often, so as to avoid "integration nightmares." The practice aims to eliminate rework and, thus, reduce cost and save time. Regular Automated Builds often deploy into a production-like environment kicking off an automated test suite to confirm that it behaves as the team expects it to.

---

Key benefits to the CAM approach include significantly higher design quality, reduced development costs and time, increased ease overcoming difficult problems, and a potentially shippable product at the end of each Sprint.

### **Different from other processes**

Traditional delivery models like a Waterfall approach often define progress by measuring in terms of deliverable artifacts such as extensive requirement specifications, design documents, test plans, etc. As contrasted to CAM, Capstone focuses on delivering high-quality working software, with artifacts in support on an as needed basis, e.g. training materials, regulatory compliance traceability, architecture diagrams, etc.

### **Measure of success**

Capstone collaborates with the Customer to define success criteria at the launch of the project, and at the end of each Sprint in the form of a Retrospective. The key success criterion for each Sprint is, "Did we do what we committed to for the Sprint?" The "retro meeting" allows the entire team to address three actions: identify what we need to stop doing, start doing, and keep doing. This simple exercise helps the team focus on what is adding value to the delivery of the project, and eliminate those activities that do not. Additional measures include: did the Product Owner get the business value envisioned?, was s/he able to change course according to business demands and get functionality delivered that makes more sense today?, was s/he able to give early feedback?, and was there a successful two-way trust relationship with the development team? As you can see, our CAM model is highly attuned to Product Owner engagement and collaboration. The ultimate measure of success for any project is working software. In CAM, we deliver potentially shippable software at the end of each Sprint.

### **Characteristics, Roles and Responsibilities**

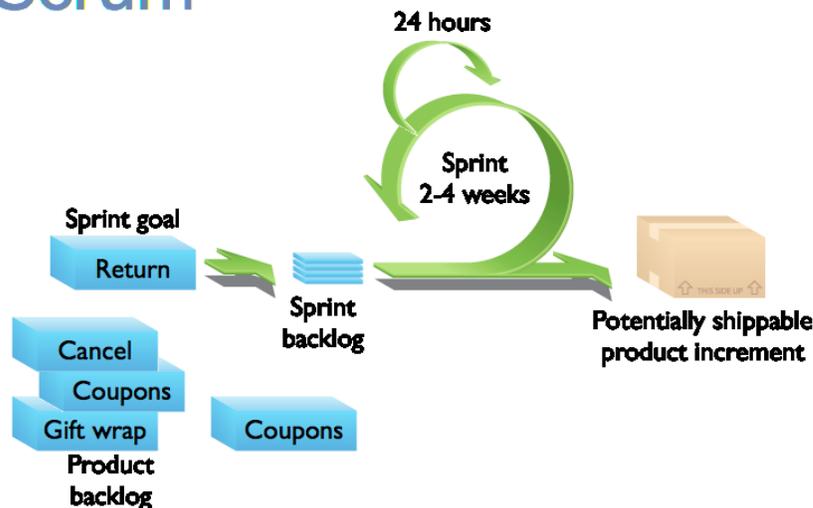
CAM's Scrum is a "process skeleton" that contains sets of practices and predefined roles. The main roles in Scrum are:

1. ScrumMaster—who maintains the processes (typically in lieu of a project manager)
2. Product Owner—who represents the stakeholders and represents the business
3. Team—a cross-functional group of about 7 people who do the actual analysis, design, implementation, testing, etc.

During each "Sprint", typically a two- to four-week period (with the length being decided by the team), the team creates a potentially shippable product increment (for example, working and tested software). The set of features that go into a Sprint come from the product "backlog," which is a prioritized set of high-level requirements of work to be done. Which backlog items go into the Sprint is determined during the Sprint Planning Meeting. During this meeting, the Product Owner informs the team of the items in the product backlog that he or she wants completed. The team then determines how much of this it can commit to complete during the next Sprint. During a Sprint, no one is allowed to change the Sprint Backlog, which means that the requirements are frozen for that Sprint. After a Sprint is completed, the team demonstrates the use of the software.

---

# Scrum



**Figure 3: SCRUM – Mountain Goat Software, Licensed by Capstone Consulting**

Scrum enables the creation of self-organizing teams by encouraging co-location of all team members, and verbal communication across all team members and disciplines that are involved in the project.

A key principle of Scrum is its recognition that during a project the customers can change their minds about what they want and need (often called requirements churn), and that unpredicted challenges cannot be easily addressed in a traditional predictive or planned manner. As such, Scrum adopts an empirical approach—accepting that the problem cannot be fully understood or defined, focusing instead on maximizing the team’s ability to deliver quickly and respond to emerging requirements.

There are several implementations of systems for managing the Scrum process, which range from 3x5 cards and whiteboards, to software packages. One of Scrum’s biggest advantages is that it is very easy to learn and requires little effort to start using.

---

## ScrumMaster (or Facilitator)

Scrum is facilitated by a ScrumMaster, whose primary job is to remove impediments to the ability of the team to deliver the Sprint goal/deliverables. The ScrumMaster is not the leader of the team (as the team is self-organizing) but acts as a buffer between the team and any distracting influences. The ScrumMaster ensures that the Scrum process is used as intended and is enforcer of rules. A key part of the ScrumMaster's role is to protect the team and keep them focused on the tasks in hand.



**Scrum Master**

- Ensures that the team is fully functional and productive
- Enables close cooperation across all roles and functions and removes barriers
- Shields the team from external interferences
- Ensures that the process is followed. Participates in daily scrum, sprint review and planning meetings

**Figure 4: ScrumMaster – Mountain Goat Software, Licensed by Capstone Consulting**

## Team

The team has the responsibility to deliver the product. A team is typically made up of 5–9 people with cross-functional skills who do the actual work (design, develop, test, technical communication, etc.).



**Team**

- Cross-functional, five to nine people
- Selects the sprint backlog
- Has the right to do anything necessary to reach the iteration goal
- Organizes itself and its work
- Demos work results to the Product Owner

**Figure 5: Team – Mountain Goat Software, Licensed by Capstone Consulting**

## Product Owner

The Product Owner represents the voice of the customer. He/she ensures that the Scrum Team works on the “right things” from a business perspective. The Product Owner writes customer-centric items (typically user stories), prioritizes them and then places them in the product backlog. A Product Owner can be a member of the Scrum Team but cannot be a ScrumMaster.

---



**Product Owner**

- Defines the features of the product, decides on release date and content
- Is responsible for the profitability of the product (ROI)
- Prioritizes features according to market value
- Can change features and priority every sprint
- Accepts or rejects work results

**Figure 6: Product Owner– Mountain Goat Software, Licensed by Capstone Consulting**

### Daily Scrum

Each day during the Sprint, a project status meeting occurs. This is called a “daily scrum,” or “the daily standup.” This meeting has specific guidelines:

- ✓ The meeting starts precisely on time
- ✓ All are welcome, but only committed team members may speak
- ✓ The meeting is time boxed to 15 minutes
- ✓ The meeting should happen at the same location and same time every day
- ✓ The meeting is conducted in front of the project board
- ✓ During the meeting, each team member answers three questions:
  - What have you done since yesterday?
  - What are you planning to do today?
  - Is there anything in your way? (It is the role of the ScrumMaster to facilitate resolution of these impediments. Typically this should occur outside the context of the daily scrum so that it may stay under 15 minutes)

### User Stories

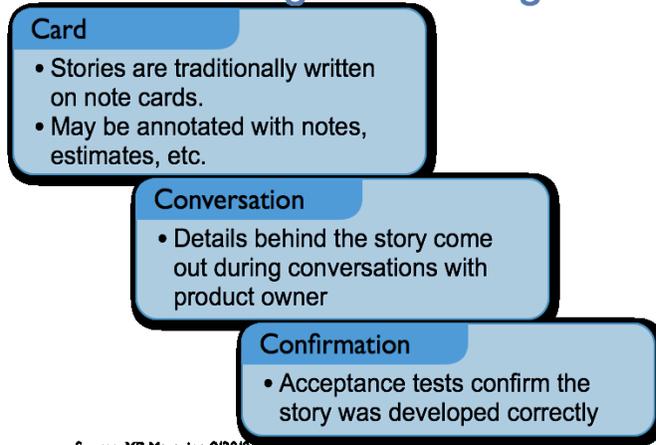
A user story is a short, simple description of a feature told from the perspective of the person who desires the new capability, usually a user or customer of the system. They typically follow a simple template:

As a <type of user>, I want <some goal> so that <some reason>.

User stories are often written on index cards or sticky notes, stored in a shoe box, and arranged on walls or tables to facilitate planning and discussion. As such, they strongly shift the focus from writing about features to discussing them. In fact, these discussions are more important than whatever text is written.

---

## Stories make great backlog items



**Figure 7: User Stories – XP Magazine**

One of the benefits is that they can be written at varying levels of detail. We can write user stories that cover large amounts of functionality. These large user stories are generally known as epics. Here is an example epic from a desktop backup product:

- ✓ As a user, I want to back up my entire hard drive

Because an epic is generally too large for an agile team to complete in one Sprint, it is split into multiple smaller stories before it is worked on. The epic above could be split into dozens (or possibly hundreds), including these two:

- ✓ As a power user, I can specify files or folders to backup based on file size, date created, and date modified
- ✓ As a user, I can indicate folders not to backup so that my backup drive isn't filled up with things I don't need saved

### Who writes User Stories?

Anyone can write a story. It's the product owner's responsibility to make sure a product backlog comprised of user stories exists, but that does not mean that the product owner is the one who writes them. Over the course of a good agile project, you should expect to have some written by each team member.

Also, note that who writes a user story is far less important than who is involved in the discussions of it.

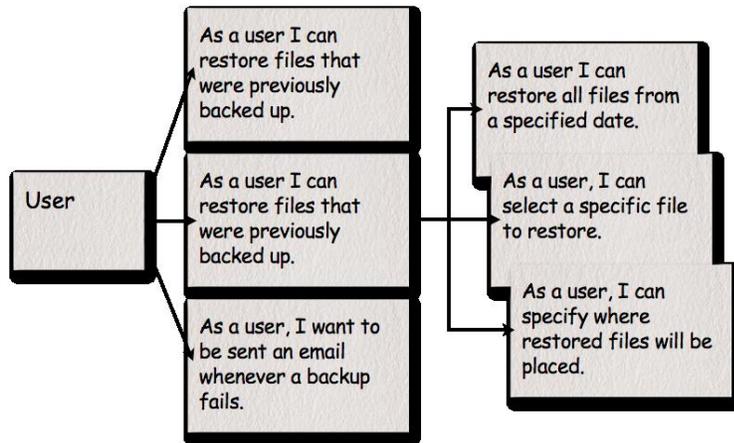
### When are Stories written?

Stories are written throughout the project. Usually a story-writing workshop is held near the start of the project. Everyone on the team participates with the goal of creating a product backlog that fully describes the functionality to be added over the course of the project or a three- to six-month release cycle within it.

Some of these stories will undoubtedly be epics. Epics will later be decomposed into smaller stories that fit more readily into a single Sprint. Additionally, new stories can be written and added to the product backlog at any time and by anyone.

---

## Start with epics and iterate



**Figure 8: Epics – Mountain Goat Software, Licensed by Capstone Consulting**

### Do Stories replace a requirements document?

Agile projects, especially Scrum ones, use a product backlog, which is a prioritized list of the functionality to be developed in a product or service. Although product backlog items can be whatever the team desires, user stories have emerged as the best and most popular form of product backlog item.

While a product backlog can be thought of as a replacement for the requirements document of a traditional project, it is important to remember that the written part of a user story (“As a user, I want ...”) is incomplete until the discussions about that story occur. It is often best to think of the written part as a pointer to the real requirement. A user story could point to a diagram depicting a workflow, a spreadsheet showing how to perform a calculation, or any other artifact the product owner or team desires.

### Planning Poker

Planning Poker is a consensus-based approach to agile estimating. To start an estimating session, the product owner or customer reads a user story or describes a feature to the estimators, who should include everyone on the team. Each estimator is holding a deck of cards with values like 0, 1, 2, 3, 5, 8, 13, 20, 40, and 100, which is the sequence we recommend. The values represent the number of story points, ideal days, or other unit in which the team estimates.

The estimators discuss the feature, asking questions of the product owner as needed. When the feature has been fully discussed, each estimator privately selects one card to represent their estimate. All cards are then revealed at the same time. If all estimators selected the same value, that becomes the estimate. If not, the estimators discuss their estimates. The high and low estimators should especially share their reasons. After further discussion, each estimator reselects an estimate card and all cards are again revealed at the same time.

The process is repeated until consensus is achieved or until the estimators decide that estimating of a particular item needs to be deferred until additional information can be acquired.

---

# Planning Poker®



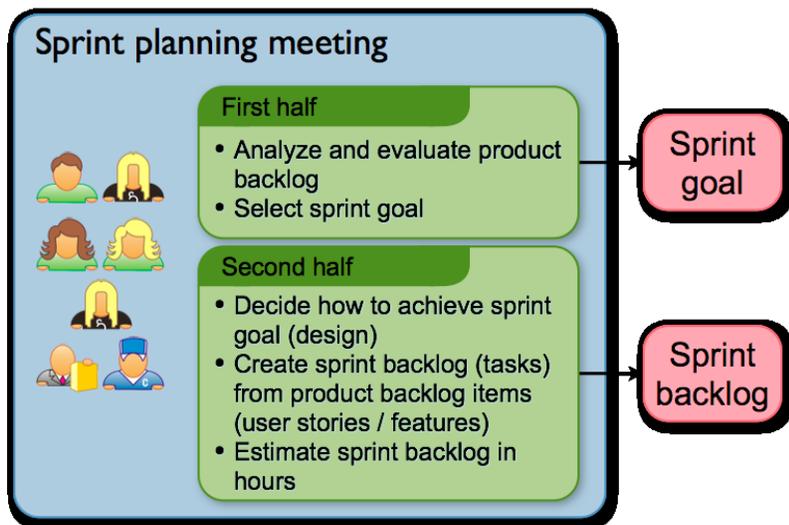
- **An iterative approach to estimating**
- **Steps**
  - **Each estimator is given a deck of cards, each card has a valid estimate written on it**
  - **Customer/Product owner reads a story and it's discussed briefly**
  - **Each estimator selects a card that's his or her estimate**
  - **Cards are turned over so all can see them**
  - **Discuss differences (especially outliers)**
  - **Re-estimate until estimates converge**

**Figure 9: Planning Poker – Mountain Goat Software, Licensed by Capstone Consulting**

## **Sprint Planning Meeting**

At the beginning of the Sprint cycle (every 7–30 days), a Sprint Planning Meeting is held

- ✓ Select what work is to be done
- ✓ Prepare the Sprint Backlog which details the time it will take to do the work
- ✓ Identify and communicate how much of the work is likely to be done during the current Sprint
- ✓ Product owner and team dialog for prioritizing the Product Backlog
- ✓ Team only hashes out a plan for the Sprint, resulting in the Sprint Backlog
- ✓ At the end of a Sprint cycle, two meetings are held: the Sprint Review Meeting and the Sprint Retrospective



**Figure 10: Planning Poker – Mountain Goat Software, Licensed by Capstone Consulting**

### **Sprint Review Meeting**

- ✓ Review the work that was completed
- ✓ Present the completed work to the stakeholders (a.k.a. "the demo").

### **Sprint Retrospective**

- ✓ All team members reflect on the past Sprint
- ✓ Identify what we want to stop doing.
- ✓ ... start doing.
- ✓ ... keep doing.
- ✓ Make continuous process improvements.

### **Information Radiators**

Information radiators are simply displays of critical team information that are continuously updated and located such that the team can see it constantly. The term "information radiator" was introduced extensively with a solid theoretical framework in Agile Software Development by Alistair Cockburn. Information radiators are sometimes referred to as Big Visible Charts.

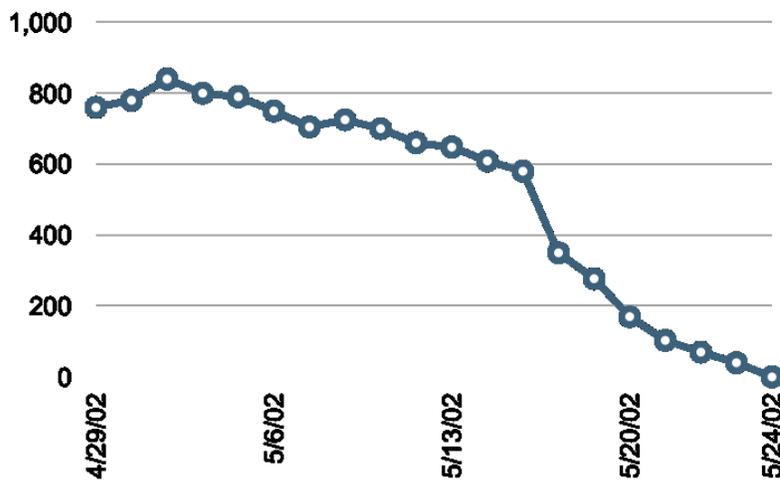
Examples of information radiators that Capstone advocates using are as follows:

- ✓ Task Boards – containing the following 3 items
  - Sprint Backlog – prioritized list of tasks to be completed during a Sprint
  - Sprint Burndown Chart – daily progress chart for a Sprint
  - Release Burndown Chart – progress chart for a release, or length of entire project

- 
- ✓ Product Backlog – a prioritized list of high level requirements or epics
  - ✓ Continuous Integration Indicator – displays the health of the build
  - ✓ Automated Test Repository – displays the number of tests running green or red

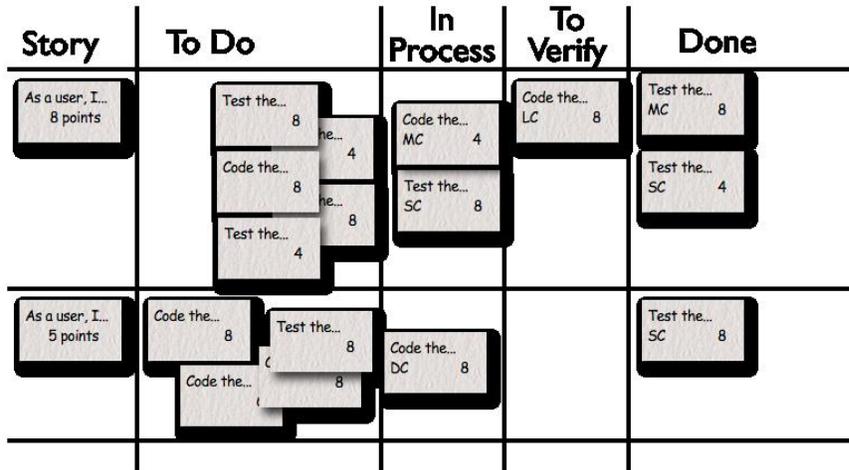
All of the information will be on prominent display at the project work site to keep the entire team aware of project health and status. Additionally, Capstone will utilize the above mentioned collaboration tools to provide visibility to the Product Owners for all of the items detailed above.

## A sprint burndown chart



**Figure 11: Sprint burndown chart – Mountain Goat Software, Licensed by Capstone Consulting**

# Task boards



**Figure 12: Task board – Mountain Goat Software, Licensed by Capstone Consulting**

## Software and Installation Quality

The CAM provides production-ready software at the end of each Sprint. CAM’s ultimate measure of success is high-quality software that the business can utilize. Quality is checked each Sprint and is a key part of the CAM.