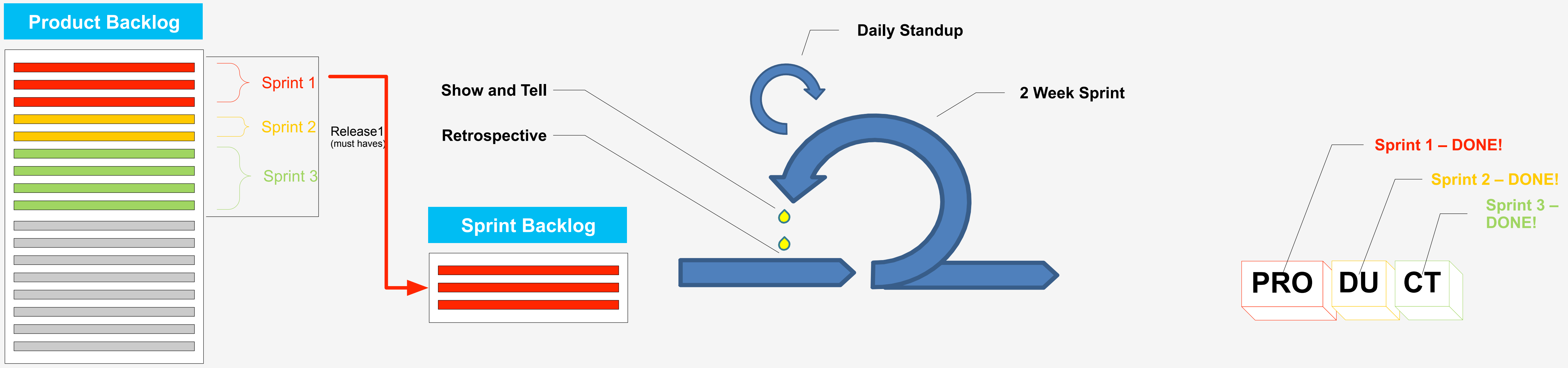
Help

Tips

Agile SDLC



Overview

Population of enough of the backlog to support the business case. The Product backlog is continually groomed through out the project

Sprint planning meeting is held to populate the sprint backlog

Sprints are 2 weeks long, this includes the sprint planning meeting, any spikes, all work been **DONE DONE**, Show and Tell and the Retrospective

At the end of the sprint, the work should be shippable (from a quality and done point of view) The Product Owner may feel that further sprints are required to make a functional release (this should align to the initial release plan, but may not)

Roles

Product Owner

- The single wringable neck
- Creates and maintains the Product Backlog
- Responsible for prioritizing the backlog

Scrum Master

- Runs the daily Standup
- Remove impediments
- Ensures scrum ceremonies are followed
- Runs retrospective at the end of the sprint

Team

- Cross functional 7 (+-2) people
- Ideally colloacted
- Creates and maintains the Sprint backlog
- Defines tasks to deliver User Stories

Checklist

□ Vision created by the team
□ Initial Business Case created
□ Cross functional resource meeting held
□ Agile alignment meeting
□ Initial User Story workshop
□ Release workshop
□ Cross functional team formed
□ Environments and C.I.A. tools in place
□ Definition of Done
□ Scrum Board setup

Scrum Board

Planned		In Progress	Verify	Done
User Story	task task task task task	task	task	task
User Story	task task task task task task task task task	task task		

Vision

- Do you have a vision for the project ?
- Has this been created and fully understood by the whole team ?
- Is it broad but engaging ?
- Is it concise ?

Examples:

- Elevator statements
- Product vision box
- User conference presentation
- Future press releases
- Magazine review

Team

- Do you have all the skill sets you need ? (Dev, QA,DW, BI,IS ?)
- Do you have all the subject matter experts identified ? (Legal, Compliance, Security)
- Are you all sitting together ?

User Story Workshop

- Run and facilitated by the Product Owner
  - Should start with aligning the team to the vision
  - Should have representation from all skill sets that will form the team

Grooming the backlog

- An ongoing process, as the team and PO go through the project as more information is acquired
- Can be a formal meeting
- Stories are reviewed, added removed from the backlog
- Team attends, run by the PO.
- Remember this may take place in the sprint timebox
- You will have more information tomorrow so Leave decisions to the last responsible moment

Environments and CI & Automation

- What environments do you need ?

  Dev
  QA
  Integration
  Performance
  Staging
  Production

- Are you using the correct tools ?

  Maven
  Perforce
  Chef
  ....

< Back

Product Backlog

At the beginning of the project, the product owner prepares a list of customer requirements prioritized by business value. This list is the Product Backlog, a single list of features prioritized by value delivered to the customer. The Scrum Team contributes to the product backlog by estimating the cost of developing features.

The Product Backlog should include all features visible to the customer, as well as the technical requirements needed to build the product. The highest priority items in the Product Backlog need to be broken down into small enough chunks to be estimable and testable. About ten developer-days of work is a good size for a Product Backlog item that can be ready for implementation in the next iteration. Features that will be implemented further out in time can be less detailed.

< Back to home

< Back to glossary

Sprint Backlog

Defines the work for a sprint, represented by the set of tasks that must be completed to realize the sprint's goals, and selected set of product backlog items.

< Back to home

< Back to glossary

Show and Tell

The show and tell is the teams opportunity to demonstrate the work they have **DONE** in the sprint.

Anyone and everyone should be invited to the meeting.

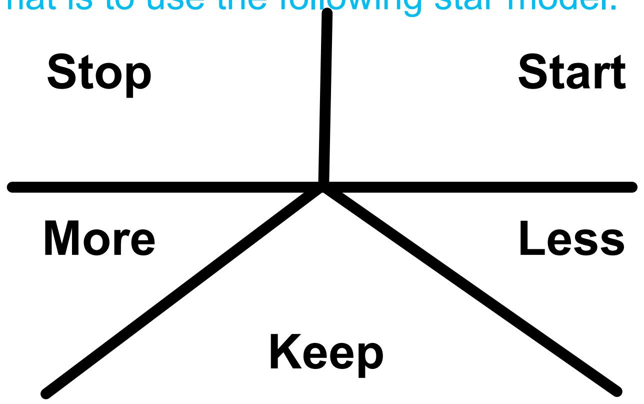
Both the Product Owner and the team undertake the presentation

< Back

Retrospective

One of the most important meetings in scrum, its part of Kaizen – Continuous improvement.

There are various methods to run your retrospective, one format is to use the following star model.



Stop

Start

More

Less

Keep

Distribute post-its to the each team member and give them 5-10 minutes to add ideas to each section, and then take turn voting on them.

< Back

Daily Standup

A daily meeting attended by all the team, including the Product owner if they wish.

Each team member should commit to coming to the meeting prepared to answer the following questions:

1. What I achieved yesterday
2. what I plan to do today
3. any impediments I have

This meeting is not a general catch up, and conversations should ideally be had after the meeting.

While the Product Owner may attend, they should not contribute to the meeting.

< Back to home

< Back to glossary

What a sprint could look like

	Mon	Tue	Wed	Thur	Fri
AM	Sprint planning	standup Sprinting	standup Sprinting	standup Sprinting	standup Sprinting
PM	Sprinting	Sprinting	Sprinting	Sprinting	Sprinting

	Mon	Tue	Wed	Thur	Fri
AM	standup Sprinting	standup Sprinting	standup Sprinting	standup Sprinting	standup Sprinting
PM	Sprinting	Sprinting	Sprinting	Sprinting	Show and tell Retospective

+ Backlog Grooming | Spikes | Release Plan review | Research

< Back

Product Owner

In Scrum, a single person must have final authority representing the customer's interest in backlog prioritization and requirements questions.

This person must be available to the team at any time, but especially during the sprint planning meeting and the sprint review meeting.

Challenges of being a product owner:

1. Resisting the temptation to "manage" the team. The team may not self-organize in the way you would expect it to. This is especially challenging if some team members request your intervention with issues the team should sort out for itself.
2. Resisting the temptation to add more important work after a Sprint is already in progress.
3. Being willing to make hard choices during the sprint planning meeting.
4. Balancing the interests of competing stakeholders.

< Back to home

< Back to glossary

Scrum Master

The ScrumMaster is a facilitator for the team and product owner. Rather than manage the team, the ScrumMaster works to assist both the team and product owner in the following ways:

* Remove the barriers between the development and the product owner so that the product owner directly drives development.
* Teach the product owner how to maximize return on investment (ROI), and meet his/her objectives through Scrum.
* Improve the lives of the development team by facilitating creativity and empowerment.
* Improve the productivity of the development team in any way possible.
* Improve the engineering practices and tools so that each increment of functionality is potentially shippable.
* Keep information about the team's progress up to date and visible to all parties.

< Back to home

< Back to glossary

Team

A team (or "Scrum team") is optimally comprised of seven plus or minus two people.

For software development projects, the team members are usually a mix of software engineers, architects, programmers, analysts, QA experts, testers, UI designers, etc. This is often called "cross-functional project teams". Agile practices also encourage cross-functional team members.

During a sprint, the team self-organizes to meet the sprint goals. The team has autonomy to choose how to best meet the goals, and is held responsible for them. The ScrumMaster acts as a guardian to ensure that the team is insulated from the product owner.

Scrum also advocates putting the entire team in one team room.

< Back to home

< Back to glossary

User Story

A good user story uses the “INVEST” model:

Independent. Reduced dependencies = easier to plan
Negotiable. Details added via collaboration
Valuable. Provides value to the customer
Estimable. Too big or too vague = not estimable
Small. Can be done in less than a week by the team
Testable. Good acceptance criteria

For more information **Click here**

< Back

Task

In Scrum, a sprint task (or task) is a unit of work generally between four and sixteen hours. Team members volunteer for tasks. They update the estimated number of hours remaining on a daily basis, influencing the sprint burndown chart. Tasks are contained by backlog items.

Scrum literature encourages splitting a task into several if the estimate exceeds twelve hours.

< Back to home

< Back to glossary

Planned

The committed user stories for the sprint.

< Back

In Progress

When the team starts to work on a task, they move into this state.

A task does not move from In progress until its **DONE**

< Back

Verify

Its important to clarify, Verify is not QA or Testing. Its a review point for the Product Owner during the sprint. Rather then waiting till the very end of the sprint, as soon as a task is in Verify, the Product Owner can review it and if there are any issues with the work, they can pass back feedback within the sprint.

< Back

Done

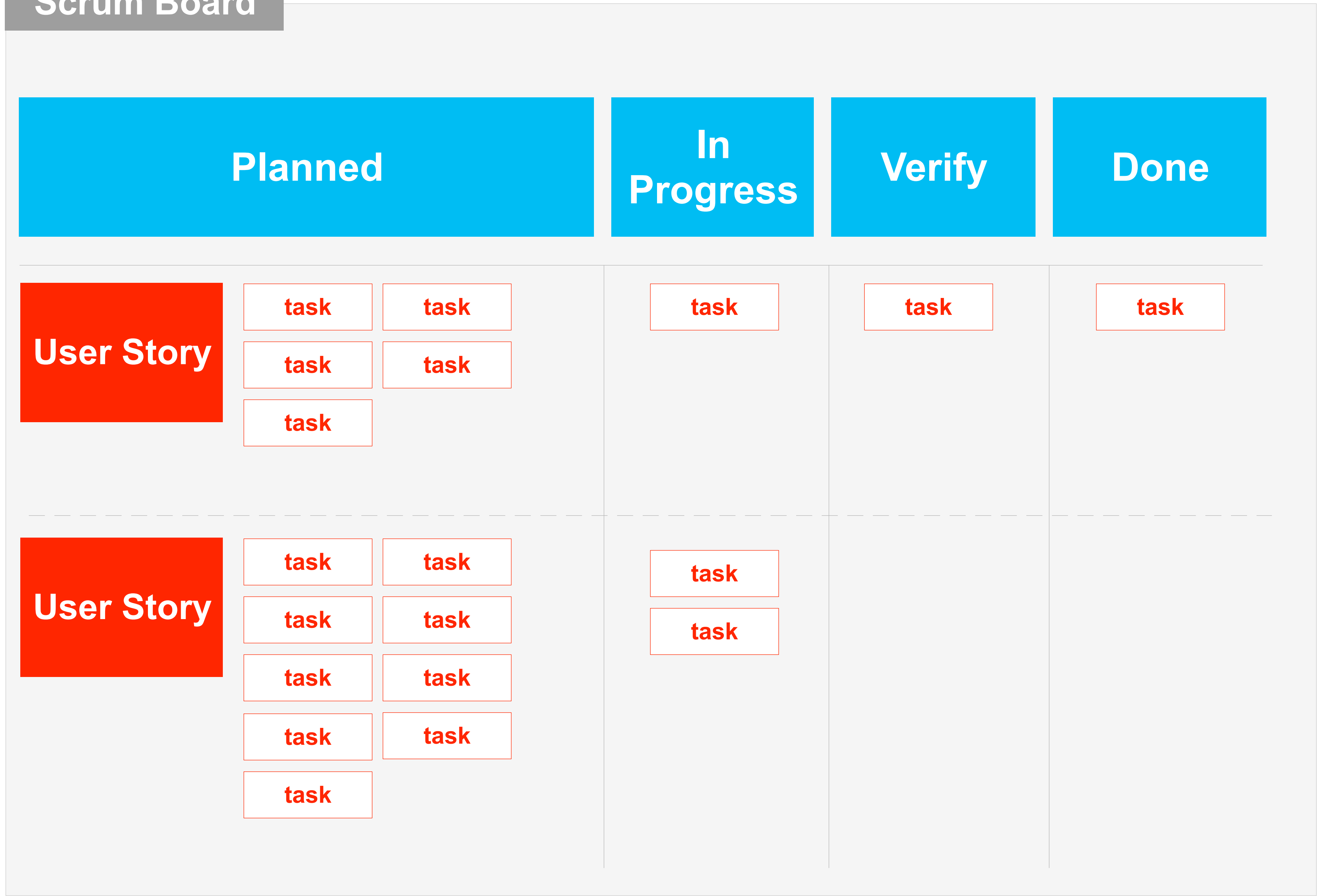
As per the team and Product Owner agreement,
Done means **Done**.

This needs to be defined for each project by the team and the product owner.

Its very likely that 2 teams could have a different definition of **Done**.

< Back

Scrum Board



Planned		In Progress	Verify	Done
User Story	task task task task task	task	task	task
User Story	task task task task task task task task task	task task		

< Back

Glossary

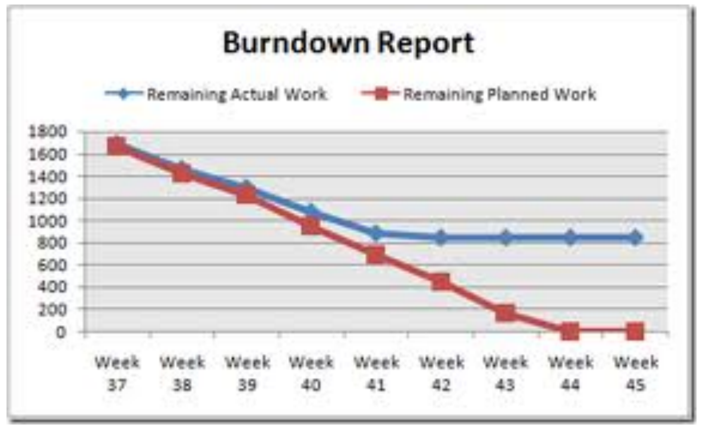
- Burndown Charts
- Daily Scrum Meeting
- Impediments
- Product Backlog
- Product Backlog Item
- Product Backlog Item Effort
- Product Burndown Chart
- Product Owner Role
- Release
- Release Burndown Chart
- Scrum Roles
- ScrumMaster Role
- Sprint
- Sprint Backlog
- Sprint Burndown Chart
- Sprint Goals
- Sprint Planning Meeting
- Sprint Retrospective Meeting
- Sprint Task
- Team
- Team Member
- Velocity

< Back

Burndown Charts

Burndown charts show work remaining over time.
Work remaining is the Y axis and time is the X axis.
The work remaining should jig up and down and eventually trend downward.

The Scrum books define a sprint burndown chart as a place to see daily progress, and a product burndown chart as where to show monthly (per sprint) progress.



< Back

Impediments

Anything that prevents a team member from performing work as efficiently as possible is an impediment. Each team member has an opportunity to announce impediments during the daily Scrum meeting. The ScrumMaster is charged with ensuring impediments get resolved. ScrumMasters often arrange sidebar meetings when impediments cannot be resolved on the spot in the daily Scrum meeting.

< Back

Product Backlog Item

In Scrum, a product backlog item ("PBI", "backlog item", or "item") is a unit of work small enough to be completed by a team in one Sprint iteration. Backlog items are decomposed into one or more tasks.

< Back

Product Backlog Item effort

Units might include story points, function points, or "t-shirt sizes" (1 for small, 2 for medium, etc.). The advantage of vaguer units is they're explicit about the distinction that product backlog item effort estimates are not estimates of duration. Also, estimates at this level are rough guesses that should never be confused with actual working hours.

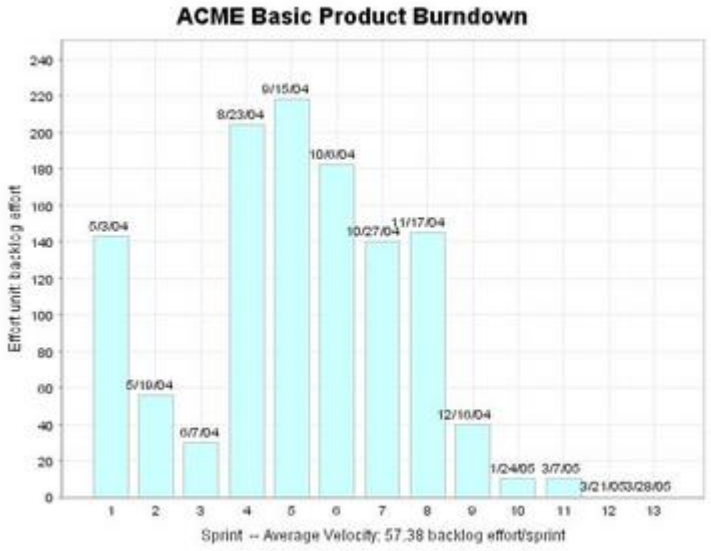
Note that sprint tasks are distinct from product backlog items and task effort remaining is always estimated in hours.

< Back

Product Backlog burndown chart

In Scrum, the product burndown chart is a "big picture" view of a project's progress. It shows how much work was left to do at the beginning of each sprint. The scope of this chart spans releases; however, a release burndown chart is limited to a single release.

The following example illustrates a product burndown chart, for an example (ACME) product:



< Back

Release

The transition of an increment of potentially shippable product from the development team into routine use by customers. Releases typically happen when one or more sprints has resulted in the product having enough value to outweigh the cost to deploy it.

"The product is released to customer or marketplace obligations. The release balances functionality, cost, and quality requirements against date commitments."

< Back

Release burn down chart

In Scrum, the release burndown chart is a "big picture" view of a release's progress. It shows how much work was left to do at the beginning of each sprint comprising a single release. The scope of this chart is a single release; however, a product burndown chart spans all releases.

< Back

Scrum Roles

Product Owner

Scrum Master

Team

< Back

Sprint

Defines the work for a sprint, represented by the set of tasks that must be completed to realize the sprint's goals, and selected set of product backlog items.

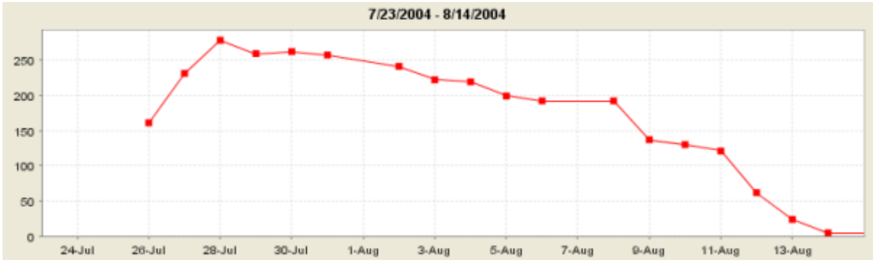
< Back

Sprint Burndown Chart

A sprint burndown chart (or "sprint burndown graph") depicts the total task hours remaining per day. This shows you where your team stands regarding completing the tasks that comprise the product backlog items that achieve the goals of the sprint. The X-axis represents days in the sprint, while the Y-axis is effort remaining (usually in ideal engineering hours).

To motivate the team, the sprint burndown chart should be displayed prominently. It also acts as an effective information radiator . A manual alternative to this is a physical task board .

Ideally the chart burns down to zero by the end of the sprint. If the team members are reporting their remaining task hours realistically, the line should bump up and down chaotically. The profile shown below is typical, and demonstrates why the "percentage done" concept of traditional project management breaks down. Assuming we started measuring on July 26, what "percentage done" were we on July 28?



< Back

Sprint Goals

Sprint goals are the result of a negotiation between the product owner and the development team.

Meaningful goals are specific and measurable. Instead of "Improve scalability" try "Handle five times as many users as version 0.8."

Scrum focuses on goals that result in demonstrable product. The product owner is entitled to expect demonstrable product (however small or flimsy) starting with the very first Sprint. In iterative development, subsequent Sprints can increase the robustness or size of the feature set.

Have your team commit to goals that anyone will be able to see are met (or not met) at the end of the sprint. At sprint review meetings, the sprint demonstration is conducted after which the team asks the product owner whether (s)he feels the goals were met.

While some specific product backlog items may not be done at the end of a sprint, it should be very unusual for a team not to meet its sprint goals. Scrum requires the team to notify the product owner as soon as it becomes aware it will not meet its goals.

< Back

Sprint Planning Meeting

The Sprint planning meeting is a negotiation between the team and the product owner about what the team will do during the next sprint.

The product owner and all team members agree on a set of sprint goals, which is used to determine which product backlog items to commit from the uncommitted backlog to the sprint. Often new backlog items are defined during the meeting. This portion of the sprint planning meeting is time-boxed to four hours.

Typically the team will then excuse the product owner from the room and break the backlog Items down into tasks. The product owner is expected to be on call during this phase (previously called the sprint definition meeting) for renegotiation or to answer questions that affect the time estimates. This portion of the sprint planning meeting is time-boxed to four hours. Sometimes teams insert placeholder tasks (with rough estimates) for the product backlog items they don't expect to start working until later in the sprint.

< Back

Team member

In Scrum parlance, a team member is defined as anyone working on sprint tasks toward the sprint goal.

< Back

Velocity

In Scrum, velocity is how much product backlog effort a team can handle in one sprint. This can be estimated by viewing previous sprints, assuming the team composition and sprint duration are kept constant. It can also be established on a sprint-by-sprint basis, using commitment-based planning.

Once established, velocity can be used to plan projects and forecast release and product completion dates.

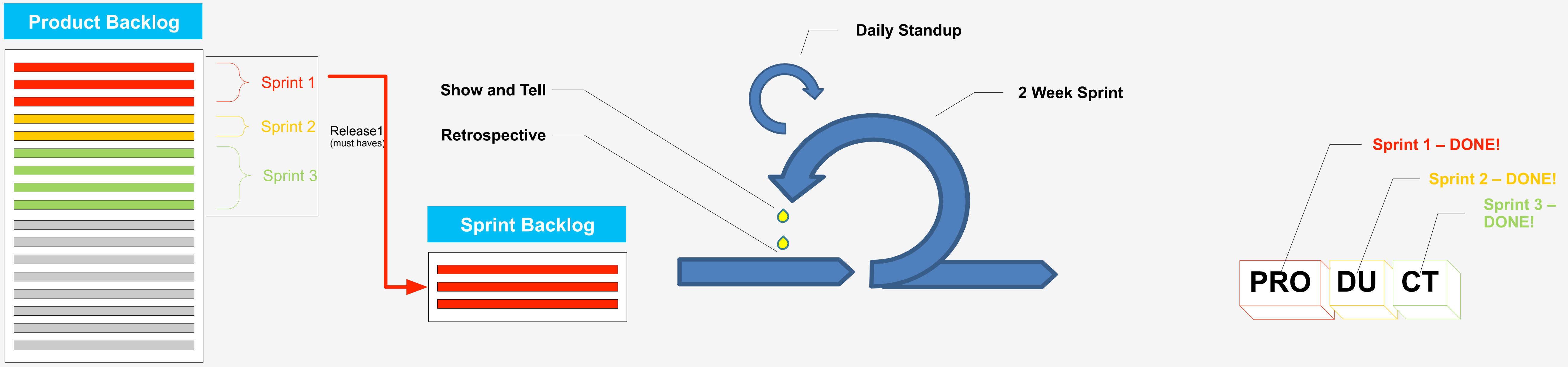
How can velocity computations be meaningful when backlog item estimates are intentionally rough? The law of large numbers tends to average out the roughness of the estimates.

< Back

Help

Tips

Agile SDLC



Overview

Population of enough of the backlog to support the business case. The Product backlog is continually groomed through out the project

Sprint planning meeting is held to populate the sprint backlog

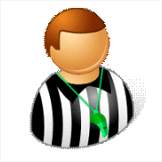
Sprints are 2 weeks long, this includes the sprint planning meeting, any spikes, all work been **DONE DONE**, Show and Tell and the Retrospective

At the end of the sprint, the work should be shippable (from a quality and done point of view) The Product Owner may feel that further sprints are required to make a functional release (this should align to the initial release plan, but may not)

Roles

Product Owner

- The single wringable neck
- Creates and maintains the Product Backlog
- Responsible for prioritizing the backlog



Scrum Master

- Runs the daily Standup
- Remove impediments
- Ensures scrum ceremonies are followed
- Runs retrospective at the end of the sprint

Team

- Cross functional 7 (+-2) people
- Ideally colloacted
- Creates and maintains the Sprint backlog
- Defines tasks to deliver User Stories

Checklist

□ Vision created by the team

□ Initial Business Case created

□ Cross functional resource meeting held

□ Agile alignment meeting

□ Initial User Story workshop

□ Release workshop

□ Cross functional team formed

□ Environments and C.I.A. tools in place

□ Definition of Done

□ Scrum Board setup

Scrum Board

Planned		In Progress	Verify	Done
User Story	task task task task task	task	task	task
User Story	task task task task task task task task task	task task		

Vision

- Do you have a vision for the project ?
- Has this been created and fully understood by the whole team ?
- Is it broad but engaging ?
- Is it concise ?

Examples:

- Elevator statements
- Product vision box
- User conference presentation
- Future press releases
- Magazine review

Team

- Do you have all the skill sets you need ? (Dev, QA,DW, BI,IS ?)
- Do you have all the subject matter experts identified ? (Legal, Compliance, Security)
- Are you all sitting together ?

User Story Workshop

- Run and facilitated by the Product Owner
  - Should start with aligning the team to the vision
  - Should have representation from all skill sets that will form the team

Grooming the backlog

- An ongoing process, as the team and PO go through the project as more information is acquired
- Can be a formal meeting
- Stories are reviewed, added removed from the backlog
- Team attends, run by the PO.
- Remember this may take place in the sprint timebox
- You will have more information tomorrow so Leave decisions to the last responsible moment

Environments and CI & Automation

- What environments do you need ?

  Dev
  QA
  Integration
  Performance
  Staging
  Production
- Are you using the correct tools ?

  Maven
  Perforce
  Chef
  ....

< Back

Product Backlog

At the beginning of the project, the product owner prepares a list of customer requirements prioritized by business value. This list is the Product Backlog, a single list of features prioritized by value delivered to the customer. The Scrum Team contributes to the product backlog by estimating the cost of developing features.

The Product Backlog should include all features visible to the customer, as well as the technical requirements needed to build the product. The highest priority items in the Product Backlog need to be broken down into small enough chunks to be estimable and testable. About ten developer-days of work is a good size for a Product Backlog item that can be ready for implementation in the next iteration. Features that will be implemented further out in time can be less detailed.

< Back to home

< Back to glossary

Sprint Backlog

Defines the work for a sprint, represented by the set of tasks that must be completed to realize the sprint's goals, and selected set of product backlog items.

< Back to home

< Back to glossary

Show and Tell

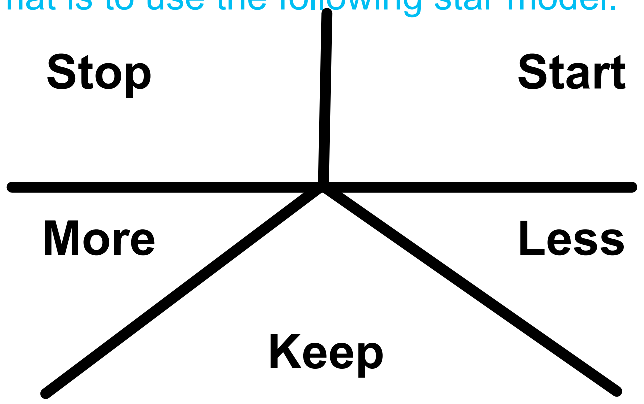
The show and tell is the teams opportunity to demonstrate the work they have **DONE** in the sprint.
Anyone and everyone should be invited to the meeting.
Both the Product Owner and the team undertake the presentation

< Back

Retrospective

One of the most important meetings in scrum, its part of Kaizen – Continuous improvement.

There are various methods to run your retrospective, one format is to use the following star model.



Distribute post-its to the each team member and give them 5-10 minutes to add ideas to each section, and then take turn voting on them.

< Back

Daily Standup

A daily meeting attended by all the team, including the Product owner if they wish.
Each team member should commit to coming to the meeting prepared to answer the following questions:

1. What I achieved yesterday
2. what I plan to do today
3. any impediments I have

This meeting is not a general catch up, and conversations should ideally be had after the meeting.
While the Product Owner may attend, they should not contribute to the meeting.

< Back to home

< Back to glossary

What a sprint could look like

	Mon	Tue	Wed	Thur	Fri
AM	Sprint planning	standup Sprinting	standup Sprinting	standup Sprinting	standup Sprinting
PM	Sprinting	Sprinting	Sprinting	Sprinting	Sprinting

	Mon	Tue	Wed	Thur	Fri
AM	standup Sprinting	standup Sprinting	standup Sprinting	standup Sprinting	standup Sprinting
PM	Sprinting	Sprinting	Sprinting	Sprinting	Show and tell Retospective

✚ Backlog Grooming | Spikes | Release Plan review | Research

< Back

Product Owner

In Scrum, a single person must have final authority representing the customer's interest in backlog prioritization and requirements questions.

This person must be available to the team at any time, but especially during the sprint planning meeting and the sprint review meeting.

Challenges of being a product owner:

1. Resisting the temptation to "manage" the team. The team may not self-organize in the way you would expect it to. This is especially challenging if some team members request your intervention with issues the team should sort out for itself.
2. Resisting the temptation to add more important work after a Sprint is already in progress.
3. Being willing to make hard choices during the sprint planning meeting.
4. Balancing the interests of competing stakeholders.

< Back to home

< Back to glossary

Scrum Master

The ScrumMaster is a facilitator for the team and product owner. Rather than manage the team, the ScrumMaster works to assist both the team and product owner in the following ways:

* Remove the barriers between the development and the product owner so that the product owner directly drives development.

* Teach the product owner how to maximize return on investment (ROI), and meet his/her objectives through Scrum.

* Improve the lives of the development team by facilitating creativity and empowerment.

* Improve the productivity of the development team in any way possible.

* Improve the engineering practices and tools so that each increment of functionality is potentially shippable.

* Keep information about the team's progress up to date and visible to all parties.

< Back to home

< Back to glossary

Team

A team (or "Scrum team") is optimally comprised of seven plus or minus two people.

For software development projects, the team members are usually a mix of software engineers, architects, programmers, analysts, QA experts, testers, UI designers, etc. This is often called "cross-functional project teams". Agile practices also encourage cross-functional team members.

During a sprint, the team self-organizes to meet the sprint goals. The team has autonomy to choose how to best meet the goals, and is held responsible for them. The ScrumMaster acts as a guardian to ensure that the team is insulated from the product owner.

Scrum also advocates putting the entire team in one team room.

< Back to home

< Back to glossary

User Story

A good user story uses the “INVEST” model:

Independent. Reduced dependencies = easier to plan
Negotiable. Details added via collaboration
Valuable. Provides value to the customer
Estimable. Too big or too vague = not estimable
Small. Can be done in less than a week by the team
Testable. Good acceptance criteria

For more information **Click here**

< Back

Task

In Scrum, a sprint task (or task) is a unit of work generally between four and sixteen hours. Team members volunteer for tasks. They update the estimated number of hours remaining on a daily basis, influencing the sprint burndown chart. Tasks are contained by backlog items.

Scrum literature encourages splitting a task into several if the estimate exceeds twelve hours.

< Back to home

< Back to glossary

Planned

The committed user stories for the sprint.

< Back

In Progress

When the team starts to work on a task, they move into this state.

A task does not move from In progress until its **DONE**

< Back

Verify

Its important to clarify, Verify is not QA or Testing. Its a review point for the Product Owner during the sprint. Rather then waiting till the very end of the sprint, as soon as a task is in Verify, the Product Owner can review it and if there are any issues with the work, they can pass back feedback within the sprint.

< Back

Done

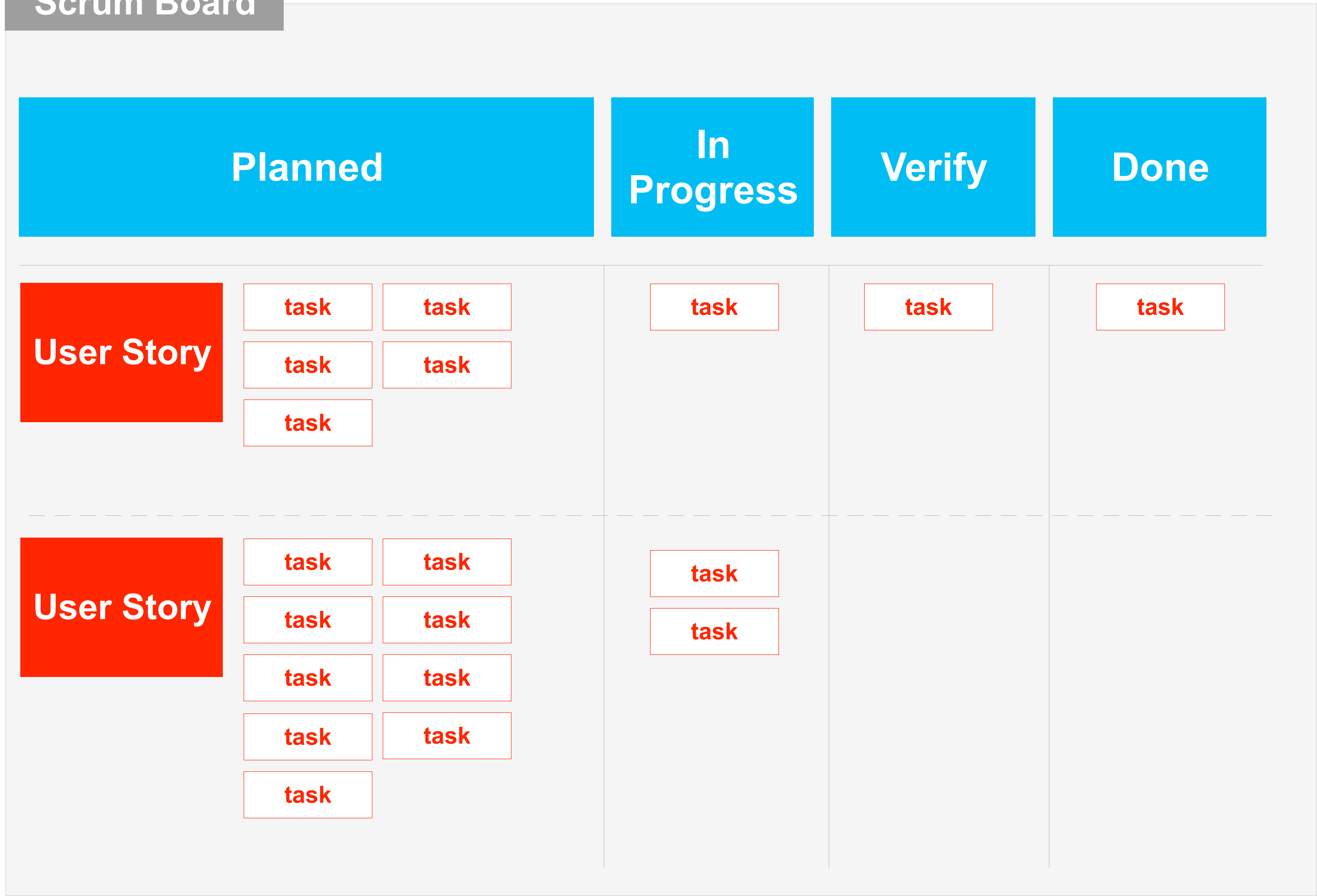
As per the team and Product Owner agreement,
Done means **Done**.

This needs to be defined for each project by the team and the product owner.

Its very likely that 2 teams could have a different definition of **Done**.

< Back

Scrum Board



	Planned	In Progress	Verify	Done
User Story	task task task task task	task	task	task
User Story	task task task task task task task task task	task task		

< Back

Glossary

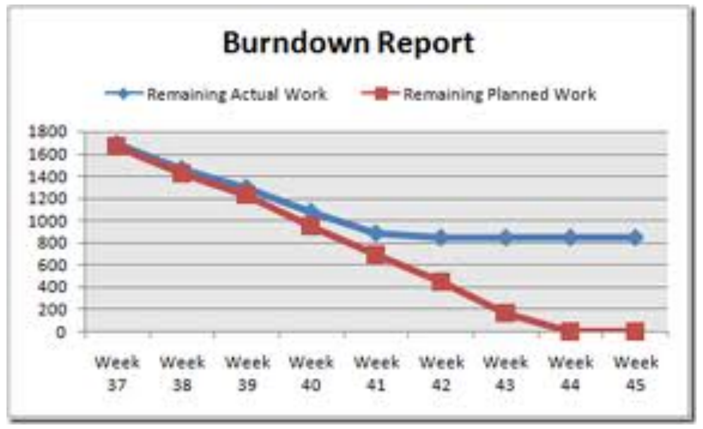
- Burndown Charts
- Daily Scrum Meeting
- Impediments
- Product Backlog
- Product Backlog Item
- Product Backlog Item Effort
- Product Burndown Chart
- Product Owner Role
- Release
- Release Burndown Chart
- Scrum Roles
- ScrumMaster Role
- Sprint
- Sprint Backlog
- Sprint Burndown Chart
- Sprint Goals
- Sprint Planning Meeting
- Sprint Retrospective Meeting
- Sprint Task
- Team
- Team Member
- Velocity

< Back

Burndown Charts

Burndown charts show work remaining over time.
Work remaining is the Y axis and time is the X axis.
The work remaining should jig up and down and eventually trend downward.

The Scrum books define a sprint burndown chart as a place to see daily progress, and a product burndown chart as where to show monthly (per sprint) progress.



< Back

Impediments

Anything that prevents a team member from performing work as efficiently as possible is an impediment. Each team member has an opportunity to announce impediments during the daily Scrum meeting. The ScrumMaster is charged with ensuring impediments get resolved. ScrumMasters often arrange sidebar meetings when impediments cannot be resolved on the spot in the daily Scrum meeting.

< Back

Product Backlog Item

In Scrum, a product backlog item ("PBI", "backlog item", or "item") is a unit of work small enough to be completed by a team in one Sprint iteration. Backlog items are decomposed into one or more tasks.

< Back

Product Backlog Item effort

Units might include story points, function points, or "t-shirt sizes" (1 for small, 2 for medium, etc.). The advantage of vaguer units is they're explicit about the distinction that product backlog item effort estimates are not estimates of duration. Also, estimates at this level are rough guesses that should never be confused with actual working hours.

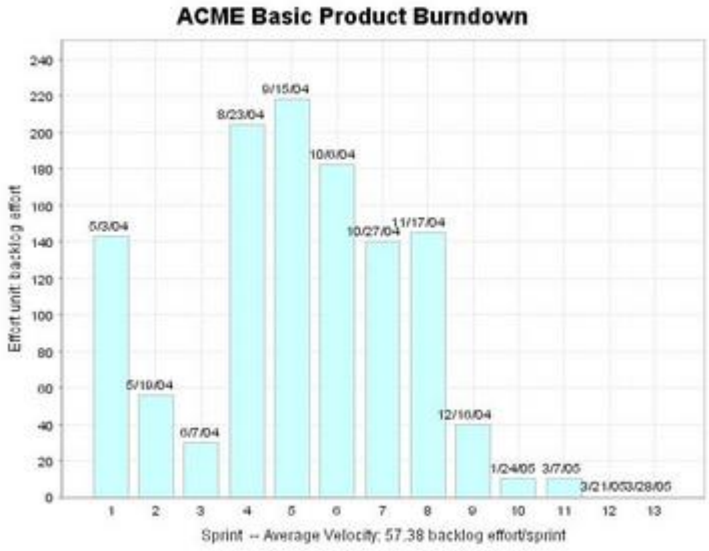
Note that sprint tasks are distinct from product backlog items and task effort remaining is always estimated in hours.

< Back

Product Backlog burndown chart

In Scrum, the product burndown chart is a "big picture" view of a project's progress. It shows how much work was left to do at the beginning of each sprint. The scope of this chart spans releases; however, a release burndown chart is limited to a single release.

The following example illustrates a product burndown chart, for an example (ACME) product:



< Back

Release

The transition of an increment of potentially shippable product from the development team into routine use by customers. Releases typically happen when one or more sprints has resulted in the product having enough value to outweigh the cost to deploy it.

"The product is released to customer or marketplace obligations. The release balances functionality, cost, and quality requirements against date commitments."

< Back

Release burn down chart

In Scrum, the release burndown chart is a "big picture" view of a release's progress. It shows how much work was left to do at the beginning of each sprint comprising a single release. The scope of this chart is a single release; however, a product burndown chart spans all releases.

< Back

Scrum Roles

Product Owner

Scrum Master

Team

< Back

Sprint

Defines the work for a sprint, represented by the set of tasks that must be completed to realize the sprint's goals, and selected set of product backlog items.

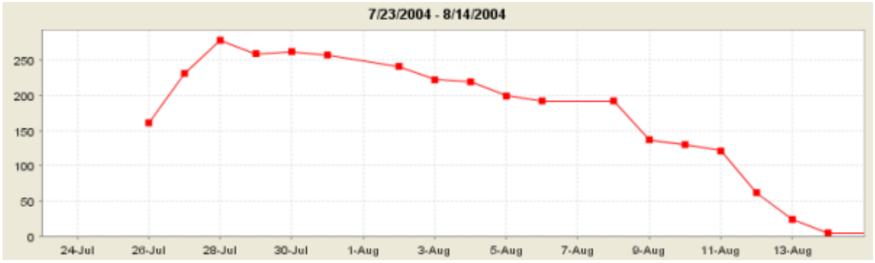
< Back

Sprint Burndown Chart

A sprint burndown chart (or "sprint burndown graph") depicts the total task hours remaining per day. This shows you where your team stands regarding completing the tasks that comprise the product backlog items that achieve the goals of the sprint. The X-axis represents days in the sprint, while the Y-axis is effort remaining (usually in ideal engineering hours).

To motivate the team, the sprint burndown chart should be displayed prominently. It also acts as an effective information radiator . A manual alternative to this is a physical task board .

Ideally the chart burns down to zero by the end of the sprint. If the team members are reporting their remaining task hours realistically, the line should bump up and down chaotically. The profile shown below is typical, and demonstrates why the "percentage done" concept of traditional project management breaks down. Assuming we started measuring on July 26, what "percentage done" were we on July 28?



< Back

Sprint Goals

Sprint goals are the result of a negotiation between the product owner and the development team.

Meaningful goals are specific and measurable. Instead of "Improve scalability" try "Handle five times as many users as version 0.8."

Scrum focuses on goals that result in demonstrable product. The product owner is entitled to expect demonstrable product (however small or flimsy) starting with the very first Sprint. In iterative development, subsequent Sprints can increase the robustness or size of the feature set.

Have your team commit to goals that anyone will be able to see are met (or not met) at the end of the sprint. At sprint review meetings, the sprint demonstration is conducted after which the team asks the product owner whether (s)he feels the goals were met.

While some specific product backlog items may not be done at the end of a sprint, it should be very unusual for a team not to meet its sprint goals. Scrum requires the team to notify the product owner as soon as it becomes aware it will not meet its goals.

< Back

Sprint Planning Meeting

The Sprint planning meeting is a negotiation between the team and the product owner about what the team will do during the next sprint.

The product owner and all team members agree on a set of sprint goals, which is used to determine which product backlog items to commit from the uncommitted backlog to the sprint. Often new backlog items are defined during the meeting. This portion of the sprint planning meeting is time-boxed to four hours.

Typically the team will then excuse the product owner from the room and break the backlog Items down into tasks. The product owner is expected to be on call during this phase (previously called the sprint definition meeting) for renegotiation or to answer questions that affect the time estimates. This portion of the sprint planning meeting is time-boxed to four hours. Sometimes teams insert placeholder tasks (with rough estimates) for the product backlog items they don't expect to start working until later in the sprint.

< Back

Team member

In Scrum parlance, a team member is defined as anyone working on sprint tasks toward the sprint goal.

< Back

Velocity

In Scrum, velocity is how much product backlog effort a team can handle in one sprint. This can be estimated by viewing previous sprints, assuming the team composition and sprint duration are kept constant. It can also be established on a sprint-by-sprint basis, using commitment-based planning.

Once established, velocity can be used to plan projects and forecast release and product completion dates.

How can velocity computations be meaningful when backlog item estimates are intentionally rough? The law of large numbers tends to average out the roughness of the estimates.

< Back

Agile and Scrum Cheat Sheet

AgileTransformation.com

Who is The ScrumMaster?

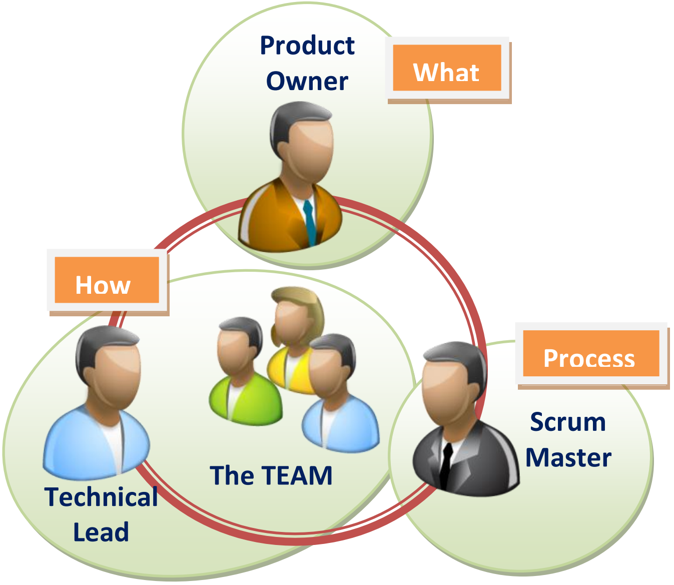
- Accountable for the **Process**
- Builds **empowered self organizing** teams
- Is a **Servant Leader**
- Removes **impediments**
- Creates **visibility, measurement, accountability** and fun
- Strong **facilitator**
- Encourages **collaboration and** open **communication**

Who is The Product Owner?

- One Person , accountable for **Backlog** and **Scope**
- **Prioritizes** the backlog
- Communicates the **Vision** of the product
- Defines **conditions of satisfaction**
- **Accepts/rejects** stories
- **Accountable** for project success
- Accountable for **ROI**
- Helps define **'Done'**
- Solicits **Customer** and **Stakeholders** feedback

Who is The Team?

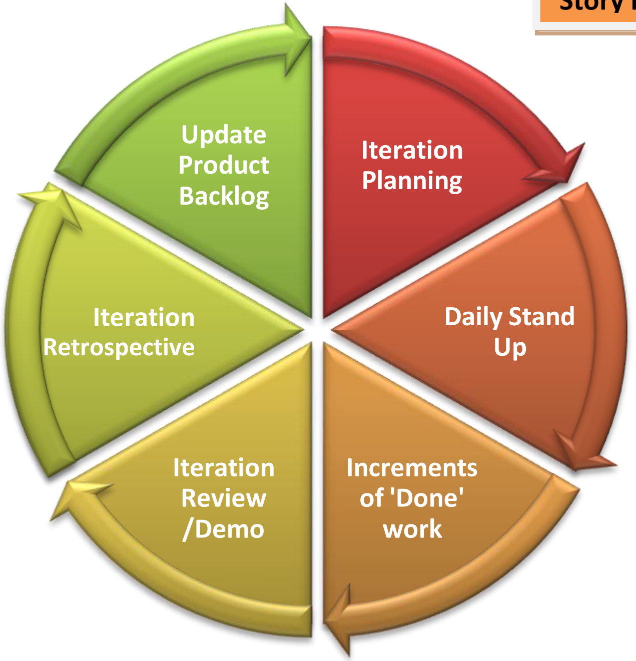
- Cross-functional team accountable for **delivering business value** iteratively
- **Engaged, Motivated, Self Organizing, Collaborative**
- **Commits** to work
- **Inspects** & **Adapts**
- **Plans** and **re-plans**
- Delivers **'Done'** stories each iteration
- Uses TDD, Automation and Best Practices



Glossary

- **BACKLOG**: one list containing all stories
- **RELEASE PLAN**: rough schedule of iterations
- **RELEASE**: Moving 'Done' stories to production
- **BURN UP CHART**: demonstrates visually how many points the team got 'Done'
- **VELOCITY:** how many points the team got 'Done' in an iteration.
- **STORY**: a description of a small valuable customer requirement
- **FEATURE/THEME**: grouping of related stories
- **TASKBOARD**: where the team tracks their tasks visibly.
- **SCRUM MASTER**: owns the process, leads the team by empowering them, removes impediments, tracks progress.
- **TEAM**: cross-functional group working together to get a story 'Done'.
- **STORY POINTS:** a relative measure of complexity for a story.
- **IMPEDIMENT**: Anything stopping progress on a task.

The Iteration/Sprint Cycle



Story Format: 'As a <role> I want to <action> so that <value>

A Story Should Be:

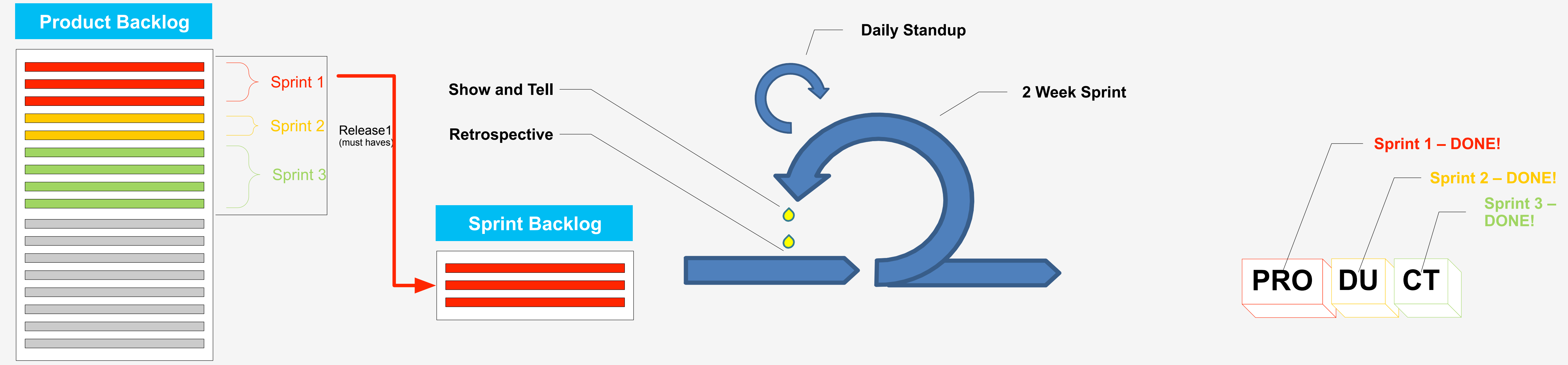
- **U**nderstandable
- **I**ndependent
- **N**egotiable
- **V**aluable
- **E**stimatable
- **S**mall
- **T**estable

Story Points

Help

Tips

Agile SDLC



Overview

Population of enough of the backlog to support the business case. The Product backlog is continually groomed through out the project

Sprint planning meeting is held to populate the sprint backlog

Sprints are 2 weeks long, this includes the sprint planning meeting, any spikes, all work been **DONE DONE**, Show and Tell and the Retrospective

At the end of the sprint, the work should be shippable (from a quality and done point of view) The Product Owner may feel that further sprints are required to make a functional release (this should align to the initial release plan, but may not)

Roles

Product Owner

- The single wringable neck
- Creates and maintains the Product Backlog
- Responsible for prioritizing the backlog

Scrum Master

- Runs the daily Standup
- Remove impediments
- Ensures scrum ceremonies are followed
- Runs retrospective at the end of the sprint

Team

- Cross functional 7 (+-2) people
- Ideally colloacted
- Creates and maintains the Sprint backlog
- Defines tasks to deliver User Stories

Checklist

□ Vision created by the team
□ Initial Business Case created
□ Cross functional resource meeting held
□ Agile alignment meeting
□ Initial User Story workshop
□ Release workshop
□ Cross functional team formed
□ Environments and C.I.A. tools in place
□ Definition of Done
□ Scrum Board setup

Scrum Board

Planned		In Progress	Verify	Done
User Story	task task task task task	task	task	task
User Story	task task task task task task task task task	task task		

Vision

- Do you have a vision for the project ?
- Has this been created and fully understood by the whole team ?
- Is it broad but engaging ?
- Is it concise ?

Examples:

- Elevator statements
- Product vision box
- User conference presentation
- Future press releases
- Magazine review

Team

- Do you have all the skill sets you need ? (Dev, QA,DW, BI,IS ?)
- Do you have all the subject matter experts identified ? (Legal, Compliance, Security)
- Are you all sitting together ?

User Story Workshop

- Run and facilitated by the Product Owner
  - Should start with aligning the team to the vision
  - Should have representation from all skill sets that will form the team

Grooming the backlog

- An ongoing process, as the team and PO go through the project as more information is acquired
- Can be a formal meeting
- Stories are reviewed, added removed from the backlog
- Team attends, run by the PO.
- Remember this may take place in the sprint timebox
- You will have more information tomorrow so Leave decisions to the last responsible moment

Environments and CI & Automation

- What environments do you need ?

  Dev
  QA
  Integration
  Performance
  Staging
  Production

- Are you using the correct tools ?

  Maven
  Perforce
  Chef
  ....

< Back

Product Backlog

At the beginning of the project, the product owner prepares a list of customer requirements prioritized by business value. This list is the Product Backlog, a single list of features prioritized by value delivered to the customer. The Scrum Team contributes to the product backlog by estimating the cost of developing features.

The Product Backlog should include all features visible to the customer, as well as the technical requirements needed to build the product. The highest priority items in the Product Backlog need to be broken down into small enough chunks to be estimable and testable. About ten developer-days of work is a good size for a Product Backlog item that can be ready for implementation in the next iteration. Features that will be implemented further out in time can be less detailed.

< Back to home

< Back to glossary

Sprint Backlog

Defines the work for a sprint, represented by the set of tasks that must be completed to realize the sprint's goals, and selected set of product backlog items.

< Back to home

< Back to glossary

Show and Tell

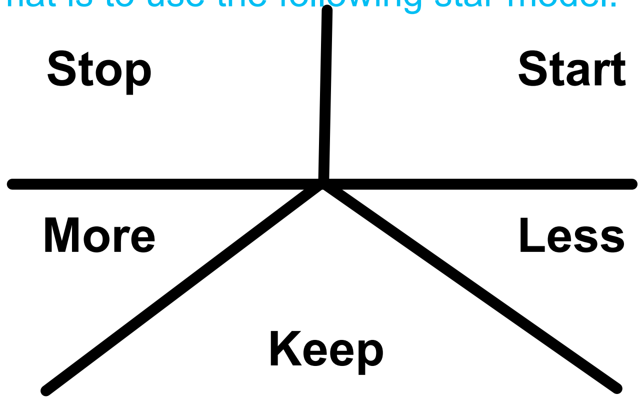
The show and tell is the teams opportunity to demonstrate the work they have **DONE** in the sprint.
Anyone and everyone should be invited to the meeting.
Both the Product Owner and the team undertake the presentation

< Back

Retrospective

One of the most important meetings in scrum, its part of Kaizen – Continuous improvement.

There are various methods to run your retrospective, one format is to use the following star model.



Stop

Start

More

Less

Keep

Distribute post-its to the each team member and give them 5-10 minutes to add ideas to each section, and then take turn voting on them.

< Back

Daily Standup

A daily meeting attended by all the team, including the Product owner if they wish.
Each team member should commit to coming to the meeting prepared to answer the following questions:

1. What I achieved yesterday
2. what I plan to do today
3. any impediments I have

This meeting is not a general catch up, and conversations should ideally be had after the meeting.
While the Product Owner may attend, they should not contribute to the meeting.

< Back to home

< Back to glossary

What a sprint could look like

	Mon	Tue	Wed	Thur	Fri
AM	Sprint planning	standup Sprinting	standup Sprinting	standup Sprinting	standup Sprinting
PM	Sprinting	Sprinting	Sprinting	Sprinting	Sprinting

	Mon	Tue	Wed	Thur	Fri
AM	standup Sprinting	standup Sprinting	standup Sprinting	standup Sprinting	standup Sprinting
PM	Sprinting	Sprinting	Sprinting	Sprinting	Show and tell Retospective

+	Backlog Grooming	Spikes	Release Plan review	Research

< Back

Product Owner

In Scrum, a single person must have final authority representing the customer's interest in backlog prioritization and requirements questions.

This person must be available to the team at any time, but especially during the sprint planning meeting and the sprint review meeting.

Challenges of being a product owner:

1. Resisting the temptation to "manage" the team. The team may not self-organize in the way you would expect it to. This is especially challenging if some team members request your intervention with issues the team should sort out for itself.
2. Resisting the temptation to add more important work after a Sprint is already in progress.
3. Being willing to make hard choices during the sprint planning meeting.
4. Balancing the interests of competing stakeholders.

< Back to home

< Back to glossary

Scrum Master

The ScrumMaster is a facilitator for the team and product owner. Rather than manage the team, the ScrumMaster works to assist both the team and product owner in the following ways:

* Remove the barriers between the development and the product owner so that the product owner directly drives development.
* Teach the product owner how to maximize return on investment (ROI), and meet his/her objectives through Scrum.
* Improve the lives of the development team by facilitating creativity and empowerment.
* Improve the productivity of the development team in any way possible.
* Improve the engineering practices and tools so that each increment of functionality is potentially shippable.
* Keep information about the team's progress up to date and visible to all parties.

< Back to home

< Back to glossary

Team

A team (or "Scrum team") is optimally comprised of seven plus or minus two people.

For software development projects, the team members are usually a mix of software engineers, architects, programmers, analysts, QA experts, testers, UI designers, etc. This is often called "cross-functional project teams". Agile practices also encourage cross-functional team members.

During a sprint, the team self-organizes to meet the sprint goals. The team has autonomy to choose how to best meet the goals, and is held responsible for them. The ScrumMaster acts as a guardian to ensure that the team is insulated from the product owner.

Scrum also advocates putting the entire team in one team room.

< Back to home

< Back to glossary

User Story

A good user story uses the “INVEST” model:

Independent. Reduced dependencies = easier to plan
Negotiable. Details added via collaboration
Valuable. Provides value to the customer
Estimable. Too big or too vague = not estimable
Small. Can be done in less than a week by the team
Testable. Good acceptance criteria

For more information **Click here**

< Back

Task

In Scrum, a sprint task (or task) is a unit of work generally between four and sixteen hours. Team members volunteer for tasks. They update the estimated number of hours remaining on a daily basis, influencing the sprint burndown chart. Tasks are contained by backlog items.

Scrum literature encourages splitting a task into several if the estimate exceeds twelve hours.

< Back to home

< Back to glossary

Planned

The committed user stories for the sprint.

< Back

In Progress

When the team starts to work on a task, they move into this state.

A task does not move from In progress until its **DONE**

< Back

Verify

Its important to clarify, Verify is not QA or Testing. Its a review point for the Product Owner during the sprint. Rather then waiting till the very end of the sprint, as soon as a task is in Verify, the Product Owner can review it and if there are any issues with the work, they can pass back feedback within the sprint.

< Back

Done

As per the team and Product Owner agreement,
Done means **Done**.

This needs to be defined for each project by the team and the product owner.

Its very likely that 2 teams could have a different definition of **Done**.

< Back

Scrum Board

Planned		In Progress	Verify	Done
User Story	task task task task task	task	task	task
User Story	task task task task task task task task task	task task		

< Back

Glossary

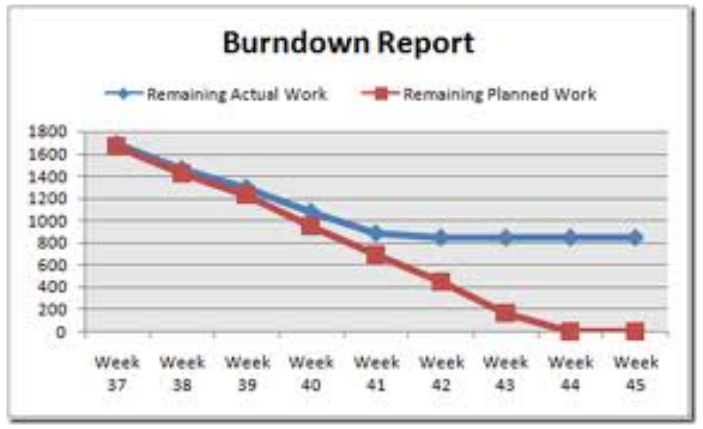
- Burndown Charts
- Daily Scrum Meeting
- Impediments
- Product Backlog
- Product Backlog Item
- Product Backlog Item Effort
- Product Burndown Chart
- Product Owner Role
- Release
- Release Burndown Chart
- Scrum Roles
- ScrumMaster Role
- Sprint
- Sprint Backlog
- Sprint Burndown Chart
- Sprint Goals
- Sprint Planning Meeting
- Sprint Retrospective Meeting
- Sprint Task
- Team
- Team Member
- Velocity

< Back

Burndown Charts

Burndown charts show work remaining over time. Work remaining is the Y axis and time is the X axis. The work remaining should jig up and down and eventually trend downward.

The Scrum books define a sprint burndown chart as a place to see daily progress, and a product burndown chart as where to show monthly (per sprint) progress.



< Back

Impediments

Anything that prevents a team member from performing work as efficiently as possible is an impediment. Each team member has an opportunity to announce impediments during the daily Scrum meeting. The ScrumMaster is charged with ensuring impediments get resolved. ScrumMasters often arrange sidebar meetings when impediments cannot be resolved on the spot in the daily Scrum meeting.

< Back

Product Backlog Item

In Scrum, a product backlog item ("PBI", "backlog item", or "item") is a unit of work small enough to be completed by a team in one Sprint iteration. Backlog items are decomposed into one or more tasks.

< Back

Product Backlog Item effort

Units might include story points, function points, or "t-shirt sizes" (1 for small, 2 for medium, etc.). The advantage of vaguer units is they're explicit about the distinction that product backlog item effort estimates are not estimates of duration. Also, estimates at this level are rough guesses that should never be confused with actual working hours.

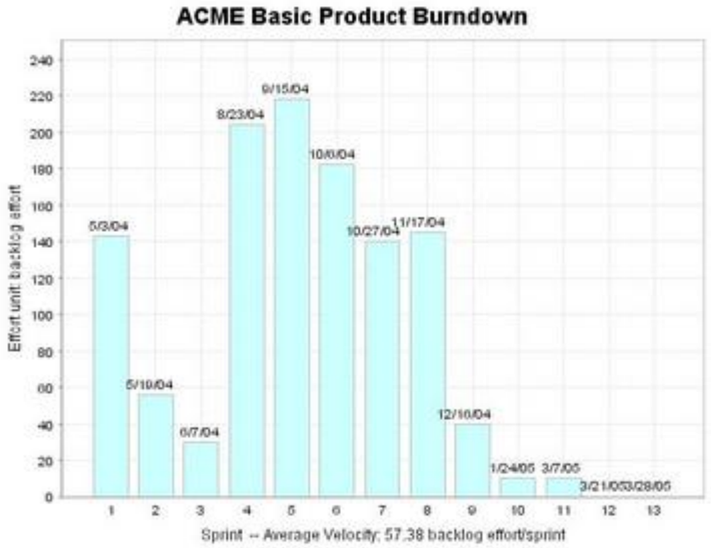
Note that sprint tasks are distinct from product backlog items and task effort remaining is always estimated in hours.

< Back

Product Backlog burndown chart

In Scrum, the product burndown chart is a "big picture" view of a project's progress. It shows how much work was left to do at the beginning of each sprint. The scope of this chart spans releases; however, a release burndown chart is limited to a single release.

The following example illustrates a product burndown chart, for an example (ACME) product:



< Back

Release

The transition of an increment of potentially shippable product from the development team into routine use by customers. Releases typically happen when one or more sprints has resulted in the product having enough value to outweigh the cost to deploy it.

"The product is released to customer or marketplace obligations. The release balances functionality, cost, and quality requirements against date commitments."

< Back

Release burn down chart

In Scrum, the release burndown chart is a "big picture" view of a release's progress. It shows how much work was left to do at the beginning of each sprint comprising a single release. The scope of this chart is a single release; however, a product burndown chart spans all releases.

< Back

Scrum Roles

Product Owner

Scrum Master

Team

< Back

Sprint

Defines the work for a sprint, represented by the set of tasks that must be completed to realize the sprint's goals, and selected set of product backlog items.

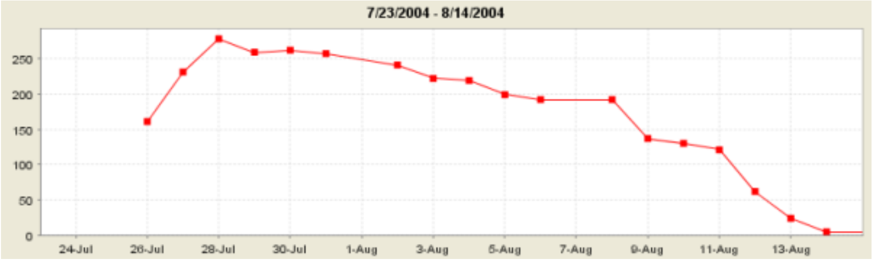
< Back

Sprint Burndown Chart

A sprint burndown chart (or "sprint burndown graph") depicts the total task hours remaining per day. This shows you where your team stands regarding completing the tasks that comprise the product backlog items that achieve the goals of the sprint. The X-axis represents days in the sprint, while the Y-axis is effort remaining (usually in ideal engineering hours).

To motivate the team, the sprint burndown chart should be displayed prominently. It also acts as an effective information radiator . A manual alternative to this is a physical task board .

Ideally the chart burns down to zero by the end of the sprint. If the team members are reporting their remaining task hours realistically, the line should bump up and down chaotically. The profile shown below is typical, and demonstrates why the "percentage done" concept of traditional project management breaks down. Assuming we started measuring on July 26, what "percentage done" were we on July 28?



< Back

Sprint Goals

Sprint goals are the result of a negotiation between the product owner and the development team.

Meaningful goals are specific and measurable. Instead of "Improve scalability" try "Handle five times as many users as version 0.8."

Scrum focuses on goals that result in demonstrable product. The product owner is entitled to expect demonstrable product (however small or flimsy) starting with the very first Sprint. In iterative development, subsequent Sprints can increase the robustness or size of the feature set.

Have your team commit to goals that anyone will be able to see are met (or not met) at the end of the sprint. At sprint review meetings, the sprint demonstration is conducted after which the team asks the product owner whether (s)he feels the goals were met.

While some specific product backlog items may not be done at the end of a sprint, it should be very unusual for a team not to meet its sprint goals. Scrum requires the team to notify the product owner as soon as it becomes aware it will not meet its goals.

< Back

Sprint Planning Meeting

The Sprint planning meeting is a negotiation between the team and the product owner about what the team will do during the next sprint.

The product owner and all team members agree on a set of sprint goals, which is used to determine which product backlog items to commit from the uncommitted backlog to the sprint. Often new backlog items are defined during the meeting. This portion of the sprint planning meeting is time-boxed to four hours.

Typically the team will then excuse the product owner from the room and break the backlog Items down into tasks. The product owner is expected to be on call during this phase (previously called the sprint definition meeting) for renegotiation or to answer questions that affect the time estimates. This portion of the sprint planning meeting is time-boxed to four hours. Sometimes teams insert placeholder tasks (with rough estimates) for the product backlog items they don't expect to start working until later in the sprint.

< Back

Team member

In Scrum parlance, a team member is defined as anyone working on sprint tasks toward the sprint goal.

< Back

Velocity

In Scrum, velocity is how much product backlog effort a team can handle in one sprint. This can be estimated by viewing previous sprints, assuming the team composition and sprint duration are kept constant. It can also be established on a sprint-by-sprint basis, using commitment-based planning.

Once established, velocity can be used to plan projects and forecast release and product completion dates.

How can velocity computations be meaningful when backlog item estimates are intentionally rough? The law of large numbers tends to average out the roughness of the estimates.

< Back

Agile and Scrum Cheat Sheet

Who is The ScrumMaster?

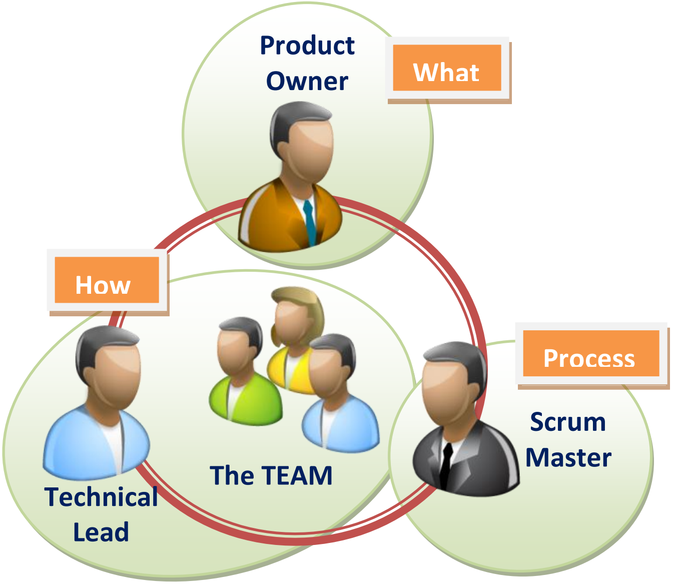
- Accountable for the **Process**
- Builds **empowered self organizing** teams
- Is a **Servant Leader**
- Removes **impediments**
- Creates **visibility, measurement, accountability** and fun
- Strong **facilitator**
- Encourages **collaboration and open communication**

Who is The Product Owner?

- One Person , accountable for **Backlog** and **Scope**
- **Prioritizes** the backlog
- Communicates the **Vision** of the product
- Defines **conditions of satisfaction**
- **Accepts/rejects** stories
- **Accountable** for project success
- Accountable for **ROI**
- Helps define **'Done'**
- Solicits **Customer** and **Stakeholders** feedback

Who is The Team?

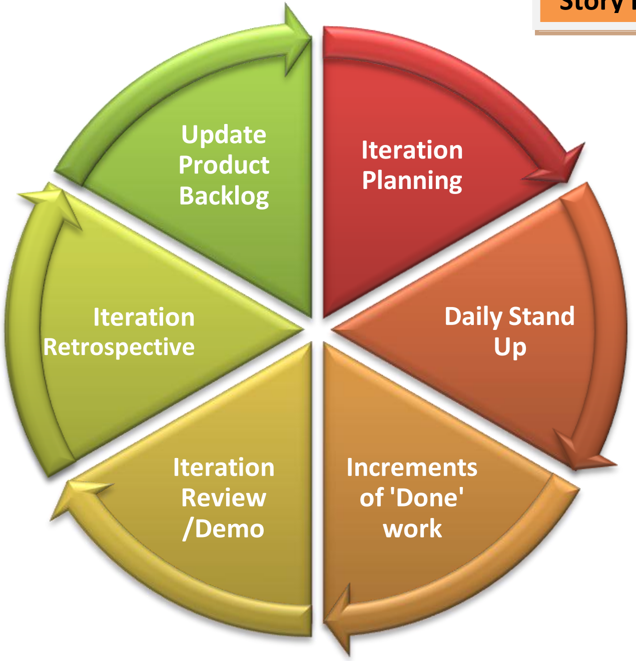
- Cross-functional team accountable for **delivering business value** iteratively
- **Engaged, Motivated, Self Organizing, Collaborative**
- **Commits** to work
- **Inspects** & **Adapts**
- **Plans** and **re-plans**
- Delivers **'Done'** stories each iteration
- Uses TDD, Automation and Best Practices



Glossary

- **BACKLOG**: one list containing all stories
- **RELEASE PLAN**: rough schedule of iterations
- **RELEASE**: Moving 'Done' stories to production
- **BURN UP CHART**: demonstrates visually how many points the team got 'Done'
- **VELOCITY:** how many points the team got 'Done' in an iteration.
- **STORY**: a description of a small valuable customer requirement
- **FEATURE/THEME**: grouping of related stories
- **TASKBOARD**: where the team tracks their tasks visibly.
- **SCRUM MASTER**: owns the process, leads the team by empowering them, removes impediments, tracks progress.
- **TEAM**: cross-functional group working together to get a story 'Done'.
- **STORY POINTS:** a relative measure of complexity for a story.
- **IMPEDIMENT**: Anything stopping progress on a task.

The Iteration/Sprint Cycle



Story Format: 'As a <role> I want to <action> so that <value>

A Story Should Be:

- **U**nderstandable
- **I**ndependent
- **N**egotiable
- **V**aluable
- **E**stimatable
- **S**mall
- **T**estable

Story Points

CONDA CHEAT SHEET

Command line package and environment manager

Learn to use conda in 30 minutes at **bit.ly/tryconda**

TIP: Anaconda Navigator is a graphical interface to use conda. Double-click the Navigator icon on your desktop or in a Terminal or at the Anaconda prompt, type `anaconda-navigator`

Conda basics

Verify conda is installed, check version number	`conda info`
Update conda to the current version	`conda update conda`
Install a package included in Anaconda	`conda install PACKAGENAME`
Run a package after install, example Spyder*	`spyder`
Update any installed program	`conda update PACKAGENAME`

Command line help	`COMMANDNAME --help`
	`conda install --help`

*Must be installed and have a deployable command, usually PACKAGENAME

Using environments

Create a new environment named py35, install Python 3.5	`conda create --name py35 python=3.5`

Activate the new environment to use it	`WINDOWS:    activate py35`
	`LINUX, macOS: source activate py35`

Get a list of all my environments, active environment is shown with *	`conda env list`
Make exact copy of an environment	`conda create --clone py35 --name py35-2`
List all packages and versions installed in active environment	`conda list`
List the history of each change to the current environment	`conda list --revisions`
Restore environment to a previous revision	`conda install --revision 2`
Save environment to a text file	`conda list --explicit > bio-env.txt`
Delete an environment and everything in it	`conda env remove --name bio-env`

Deactivate the current environment	`WINDOWS: deactivate`
	`macOS, LINUX: source deactivate`

Create environment from a text file	`conda env create --file bio-env.txt`
Stack commands: create a new environment, name it bio-env and install the biopython package	`conda create --name bio-env biopython`

Finding conda packages

Use conda to search for a package	`conda search PACKAGENAME`
See list of all packages in Anaconda	https://docs.anaconda.com/anaconda/packages/pkg-docs

CONTINUED ON BACK ➜

Installing and updating packages

Install a new package (Jupyter Notebook) in the active environment	`conda install jupyter`
Run an installed package (Jupyter Notebook)	`jupyter-notebook`
Install a new package (toolz) in a different environment (bio-env)	`conda install --name bio-env toolz`
Update a package in the current environment	`conda update scikit-learn`
Install a package (boltons) from a specific channel (conda-forge)	`conda install --channel conda-forge boltons`
Install a package directly from PyPI into the current active environment using pip	`pip install boltons`
Remove one or more packages (toolz, boltons) from a specific environment (bio-env)	`conda remove --name bio-env toolz boltons`

Managing multiple versions of Python

Install different version of Python in a new environment named py34	`conda create --name py34 python=3.4`

Switch to the new environment that has a different version of Python	`Windows:    activate py34`
	`Linux, macOS:      source activate py34`

Show the locations of all versions of Python that are currently in the path
NOTE: The first version of Python in the list will be executed.
	`Windows:    where python`
	`Linux, macOS: which -a python`

Show version information for the current active Python	`python --version`

Specifying version numbers

Ways to specify a package version number for use with conda create or conda install commands, and in meta.yaml files.

Constraint type	Specification	Result
Fuzzy	`numpy=1.11`	1.11.0, 1.11.1, 1.11.2, 1.11.18 etc.
Exact	`numpy==1.11`	1.11.0
Greater than or equal to	`"numpy>=1.11"`	1.11.0 or higher
OR	`"numpy=1.11.1|1.11.3"`	1.11.1, 1.11.3
AND	`"numpy>=1.8,<2"`	1.8, 1.9, not 2.0

NOTE: Quotation marks must be used when your specification contains a space or any of these characters: > < | *

MORE RESOURCES

Free Community Support	groups.google.com/a/continuum.io/forum/#!forum/conda
Online Documentation	conda.io/docs
Command Reference	conda.io/docs/commands
Paid Support Options	anaconda.com/support
Anaconda Onsite Training Courses	anaconda.com/training
Anaconda Consulting Services	anaconda.com/consulting

Follow us on Twitter @anacondainc and join the #AnacondaCrew!
Connect with other talented, like-minded data scientists and developers while contributing to the open source movement. Visit anaconda.com/community

ANACONDA.

anaconda.com · info@anaconda.com · 512-776-1066
8/20/2017 conda cheat sheet Version 4.3.24