Agile Development and Requirements Engineering

Unit 2

AGILE SOFTWARE DEVELOPMENT

Topics

- Introduction— Agility.
- The Agile Manifesto.
- Agile Principles.
- Other Agile Process models: like XP, Scrum, Crystal, AM, AUP.

Background study

- Features of current environment:
- Now-a-days clients do not give clear requirements so,
  - **Unclear requirement**
  - **Frequent requirement changes**
- Environment **also change which affect our project**.
  - Quickly release of project,
  - Developing software using new technology – we may not have some support document.
  - New type of project
- To handle such situation we go for **Agile model**

What Is Agile

- Agile **combines** a **philosophy** and a set of **development guidelines**.

- The philosophy encourages
  - **customer satisfaction** and **early incremental delivery of software,**
  - **Small but highly motivated project teams,**
  - **informal methods,**
  - **minimal software engineering work products,**
  - **and overall development simplicity.**

- The development guidelines stress
  - **delivery over analysis and design ,**
  - **active and continuous communication between developers and customers**

Agile Software Development

- **Agile** software development is a **conceptual framework** for software engineering that promotes development iterations **throughout the life-cycle** of the project.

- Agile methods also **emphasize working software** as the primary measure of progress.

Agile Software Development

- **Characteristics** of Agile Software Development
  - -- Light Weighted methodology
  - -- Small to medium sized teams
  - -- vague and/or changing requirements
  - -- vague and/or changing techniques
  - -- Simple design
  - -- Minimal system into production

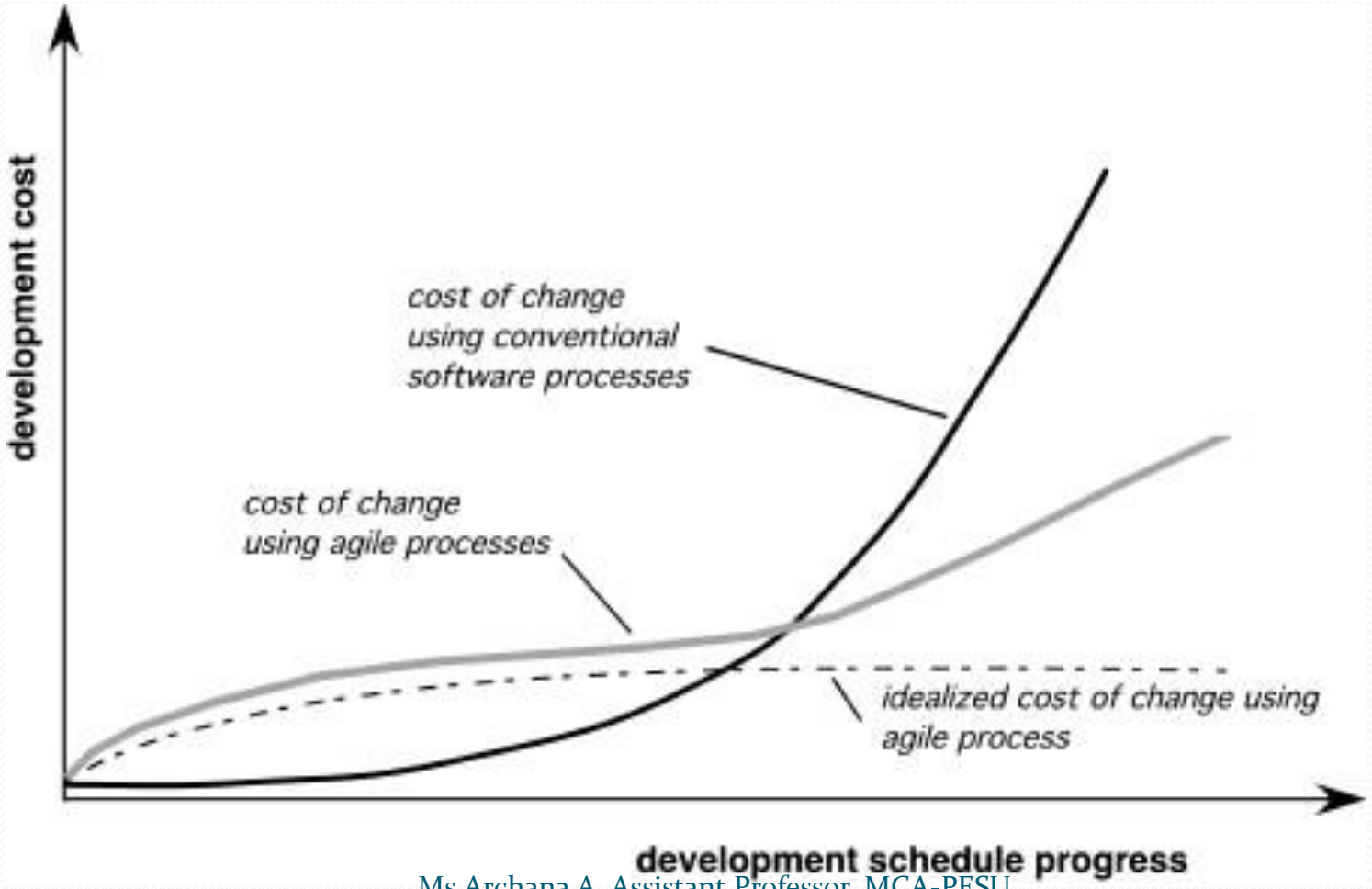
What is "Agility"?

- Effective (**rapid and adaptive**) response to change.
- Effective communication among all stakeholders.
- Drawing the customer onto the team.
- Organizing a team so that it is in control of the work performed.

Yielding …

- **Rapid, incremental delivery of software**.

Agility and the Cost of Change



An Agile Process

- Is driven **by customer descriptions** of what is required (scenarios).
- Recognizes that plans are short-lived.
- **Develops software iteratively** with a heavy emphasis on construction activities.
- Delivers **multiple** 'software increments'.
- **Adapts** as changes occur.

Agility

- **Agility is**
  - **dynamic,**
  - **content specific,**
  - **aggressively change embracing, and**
  - **growth oriented.**

Agile Principles

Agility Principles

1. Our highest priority is to **satisfy the customer** through early and continuous delivery of valuable software.

- **Customer satisfaction**
- **Early and continuous delivery**
- **Valuable software**

Agility Principles…

2. **Welcome changing requirements**, even late in development. Agile processes harness change for the customer's competitive advantage.

- **Welcome changes instead of scope creep worry**
- **Learn to accept that changes are inevitable and dynamic**

Agility Principles…

3.**Deliver working software frequently,** in a couple of weeks from a couple of months, with a preference to the shorter timescale.

- **Frequent delivery leads to frequent feedback.**
- **Frequent testing and continuous integration leads to delivery quality software.**
- **Frequent delivery helps to keep the business engaged/ active during development**.

Agility Principles …

4.Business people and developers must **work together** daily throughout the project.

- **Daily face-to-face interaction with business representative.**
- **Frequent software delivery and demo will result in frequent engagement with business.**
- **Frequent interaction will help business to fine tune their requirements based on development team suggestions.**

Agility Principles…

5. Build projects around **motivated individuals**. Give them the environment and support they need, and trust them to get the job done.

- **Motivated and smart development team will lead to deliver project successfully and efficiently.**
- **Promote empowered team. Treat them as expert and support them.**
- **Give autonomy to the team for organizing and planning their own work.**

Agility Principles…

6. The most efficient and effective method of conveying information to and within a development team **is face–to–face conversation.**

- **F2F communication is a quick mechanism to transfer information.**
- **F2F helps to convey emotions and body language.**
- **F2F helps to resolve issues immediately.**
- **Follow F2F communication where ever possible.**

Agility Principles…

7. **Working software** is the primary measure of progress.

- **Working software means tested and accepted by business.**
- **Working software means product can be used by business.**
- **Internal or partial delivery are not recognized as working software.**
- **Focus will be more on delivering working software over extensive documentation.**

Agility Principles…

8. Agile processes promote **sustainable development.** The sponsors, developers, and users should be able to maintain **a constant pace indefinitely.**

- **Constant pace supports work life balance.**
- **Happy teams get along well with business counterparts.**

Agility Principles…

9. Continuous attention to technical excellence and good design enhances agility.

- **It is important to address dynamic changes.**
- **Help to maintain change and extend the system.**

Agility Principles…

10. **Simplicity** – the art of maximizing the amount of work, not done – is essential.

- **Implement the essential elements first.**
- **Incrementally develop software focusing on Must have features.**

Agility Principles…

11. The best architectures, requirements, and designs emerge from **self–organizing teams.**

- **Allow the team to self-organize to produce better work.**
- **Self-organized team take ownership and show pride in their work.**

Agility Principles…

12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

- **Frequent retrospectives (means affecting/ influencing past things) are recommended to identify, opportunities for improvement.**
- **Helps to adjust the working behavior of the team.**

Agile Process Models

Methodologies

Extreme Programming(XP)

- **Most prominent** Agile Software development method.
- **Prescribes** a set of **daily stakeholder practices**.
- "**Extreme**" **levels** of practicing leads to more responsive software.
- Changes are more **realistic, natural**, inescapable.

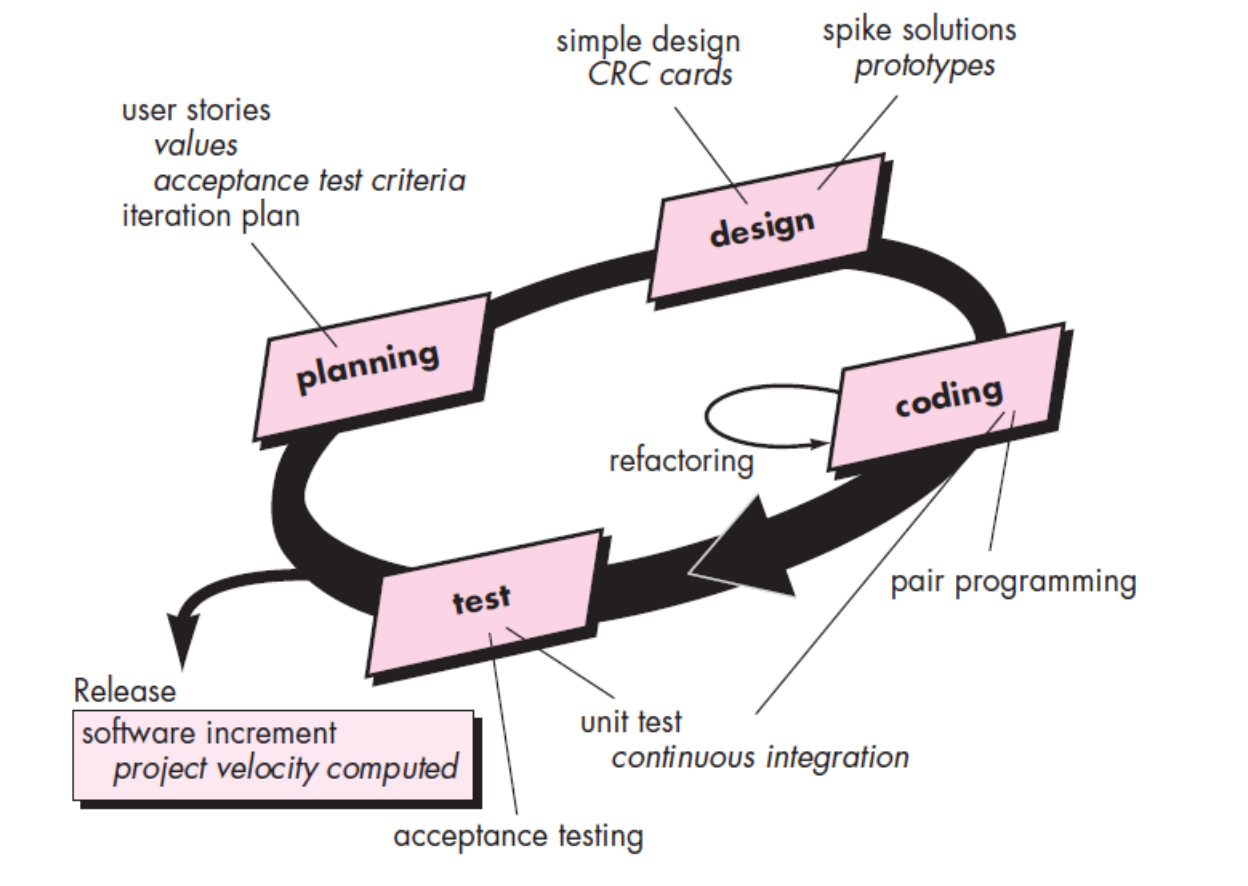
XP Values

- A set of **five values** that establish a foundation for all work performed as part of XP—
  - **communication,**
  - **simplicity,**
  - **feedback,**
  - **courage, and**
  - **respect.**
- Each of these values is used as a driver for specific XP activities, actions, and tasks.

XP Process

- The most widely used agile process, originally proposed by **Kent Beck**.
- Extreme Programming uses an **object-oriented approach.**
- Encompasses **a set of rules and practices** that occur within the context of four framework activities: **planning, design, coding, and testing.**

XP Process


simple design
CRC cards
spike solutions
prototypes
user stories
values
acceptance test criteria
iteration plan
design
planning
coding
refactoring
test
pair programming
Release
software increment
project velocity computed
unit test
continuous integration
acceptance testing

1. XP Planning:

- The planning activity *begins with* ***listening*** , Listening leads to the creation of a set of "**stories**" *that describe* ***required output,*** **features, and functionality** for software to be built.

  - Begins with the creation of "**user stories**".
  - Agile team assesses each story and assigns a **cost.**
  - Stories are grouped for a **deliverable increment.**
  - A **commitment** is made on delivery date.
  - After the first increment "**project velocity**" is used to help define subsequent delivery dates for other increments.

2. XP Design:

- Follows the **KIS principle** **(Keep It Simple).**
- Encourage the use of **CRC cards** as an **effective mechanism** for thinking about the software in an **object-oriented context**.
- CRC (class-responsibility-collaborator) cards **identify and organize the object-oriented classes** that are relevant to the current software increment.

XP Design…

- For difficult design problems, XP suggests the creation of "**spike solutions**"—a design prototype
- XP encourages "**refactoring**"—an iterative refinement of the internal program design.
  - Refactoring is a process of changing a software system in such a way that **it does not alter the external behavior** of the code, yet improves the internal structure.

3. XP Coding:

- Recommends the **construction of a unit test** for a story ***before*** **coding commences**. This helps developer to better focus on what must be implemented to pass the test.

- Encourages "**pair programming**". This provides a mechanism for real time problem solving, real time quality assurance as it is created.

- As pair programmers complete their work, the code they develop is integrated with the work of other.
- All **unit tests** are executed daily**.**

4. XP Testing:

- All **unit tests are executed daily.**
- "**Acceptance tests**" are defined by the customer and executed to assess customer visible functionality.
  - XP *acceptance tests, also called* ***customer tests****, are specified by the customer and* focus on **overall system features and functionality** that are visible and reviewable by the customer.

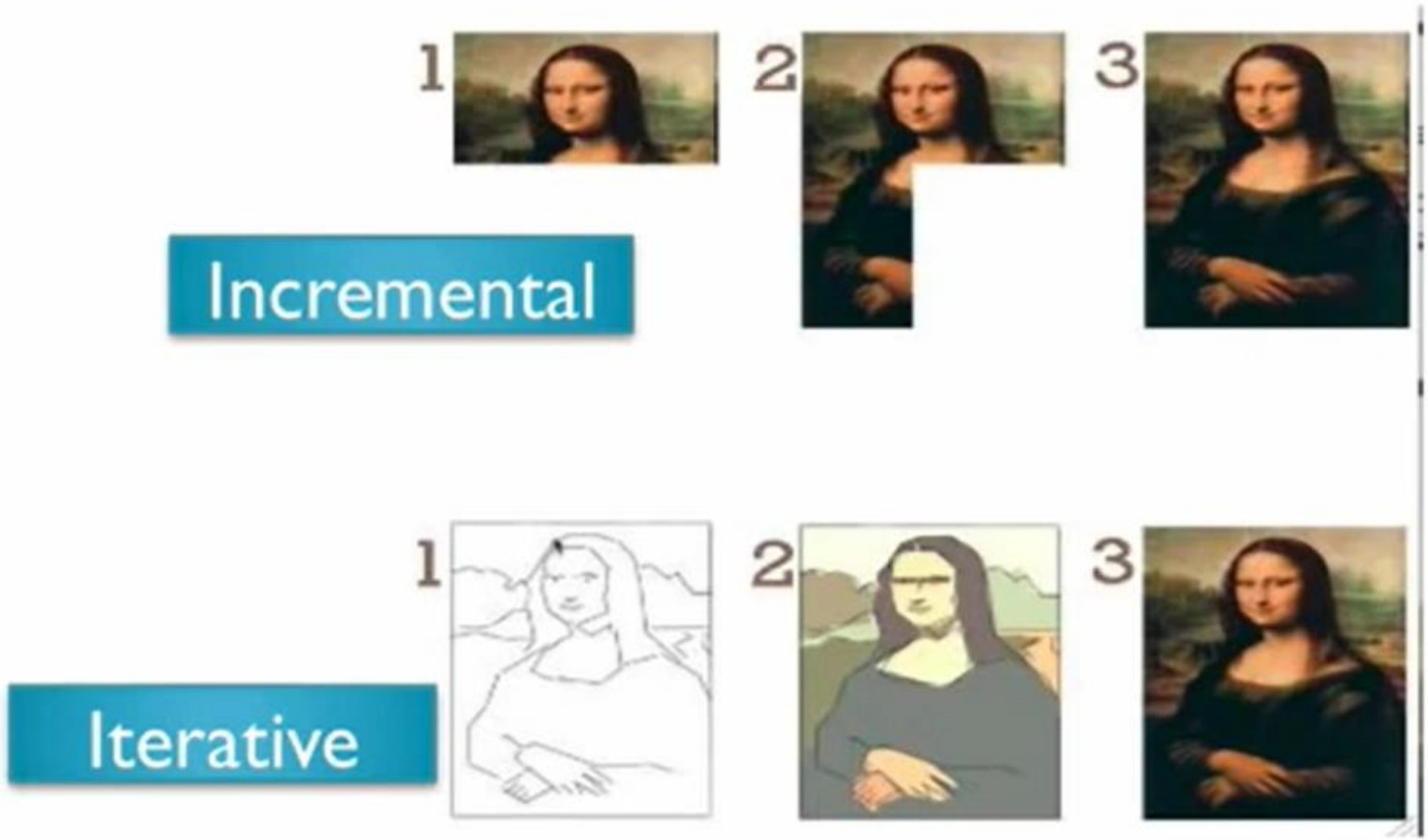
SCRUM

- Originally proposed by **Schwaber and Beedle.**

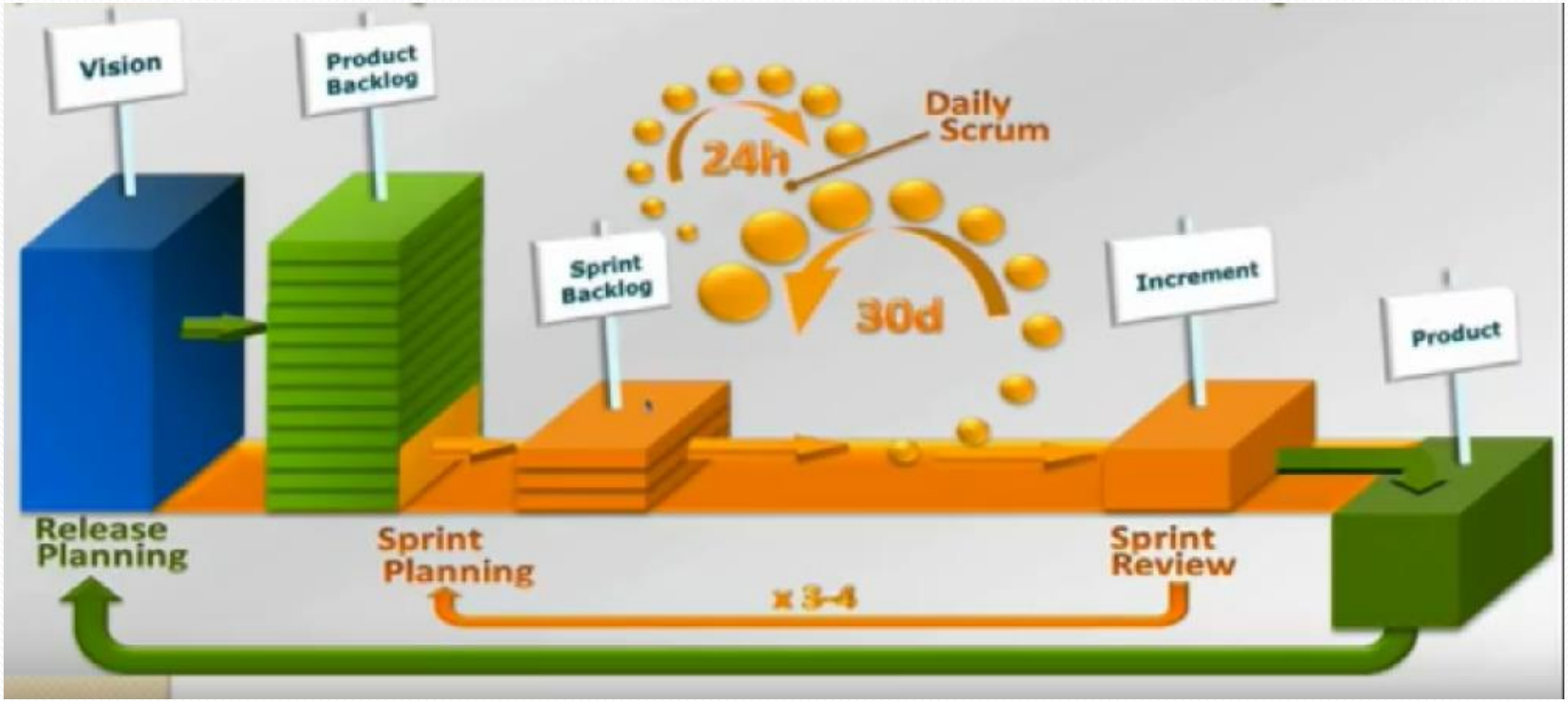
- **"Scrum** is a framework for developing complex products and systems. It is grounded in empirical process and control theory. Scrum employs an iterative and incremental approach to optimize predictability & control risk"

  – Ken Schwaber.

Iterative & Incremental approach



Scrum overview

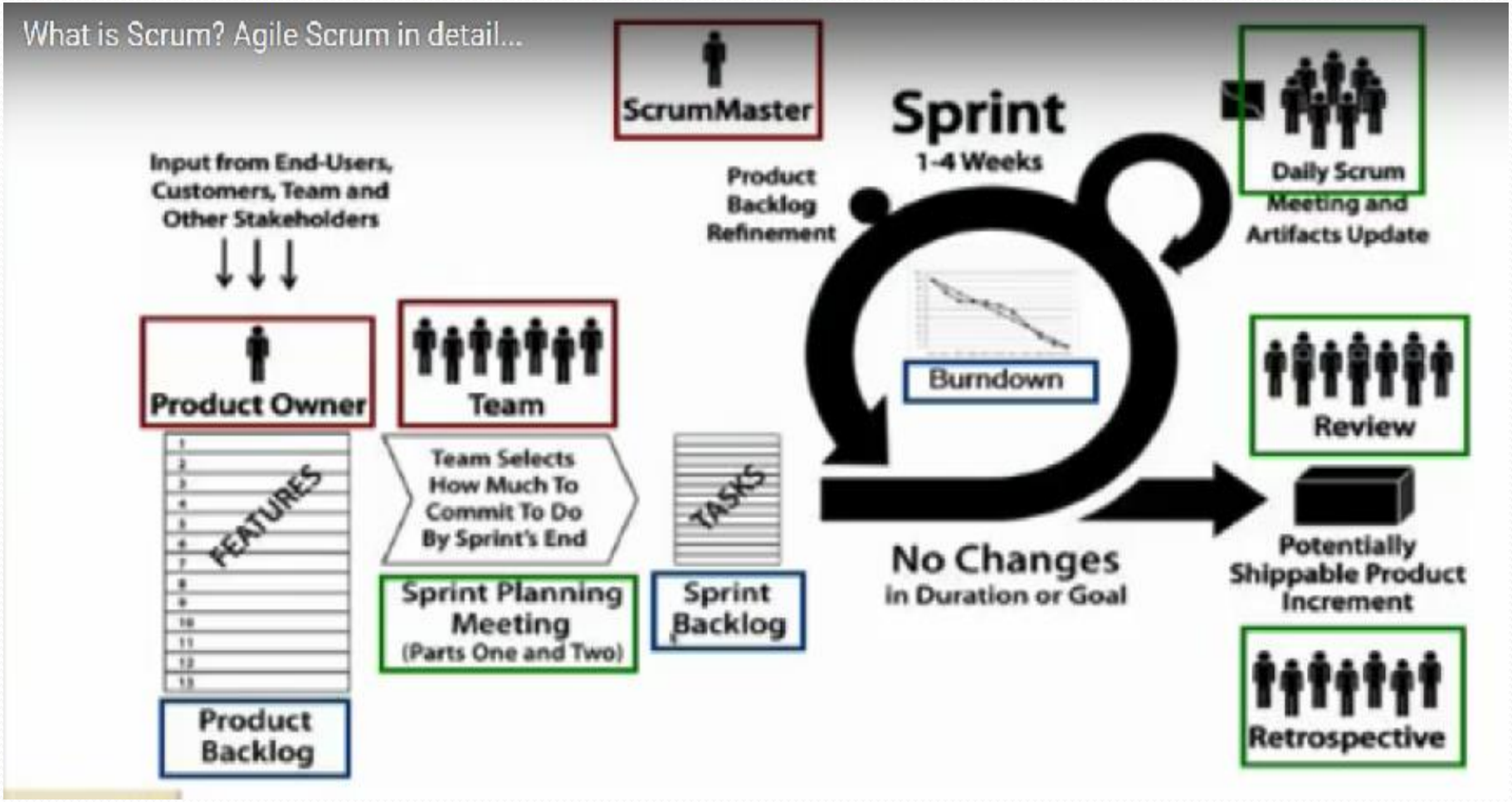


Scrum - Roles

- There are 3 Roles :

1. Product Owner
2. Team
3. Scrum Master

Scrum...


What is Scrum? Agile Scrum in detail...
ScrumMaster
Sprint
1-4 Weeks
Input from End-Users, Customers, Team and Other Stakeholders
Product Backlog Refinement
Daily Scrum Meeting and Artifacts Update
Product Owner
Team
Burndown
Review
FEATURES
Team Selects How Much To Commit To Do By Sprint's End
TASKS
Sprint Planning Meeting (Parts One and Two)
Sprint Backlog
No Changes in Duration or Goal
Potentially Shippable Product Increment
Product Backlog
Retrospective

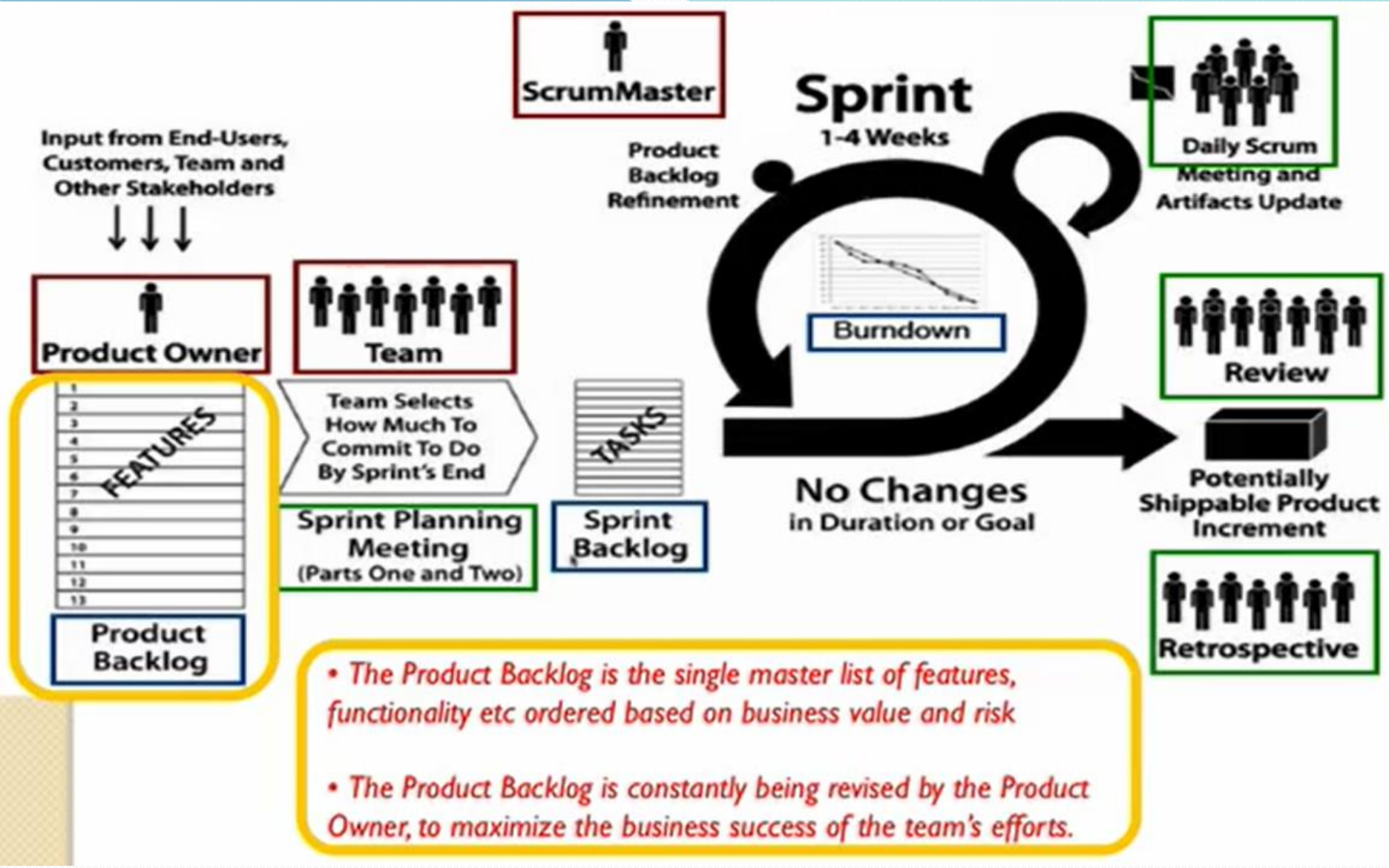
Scrum…

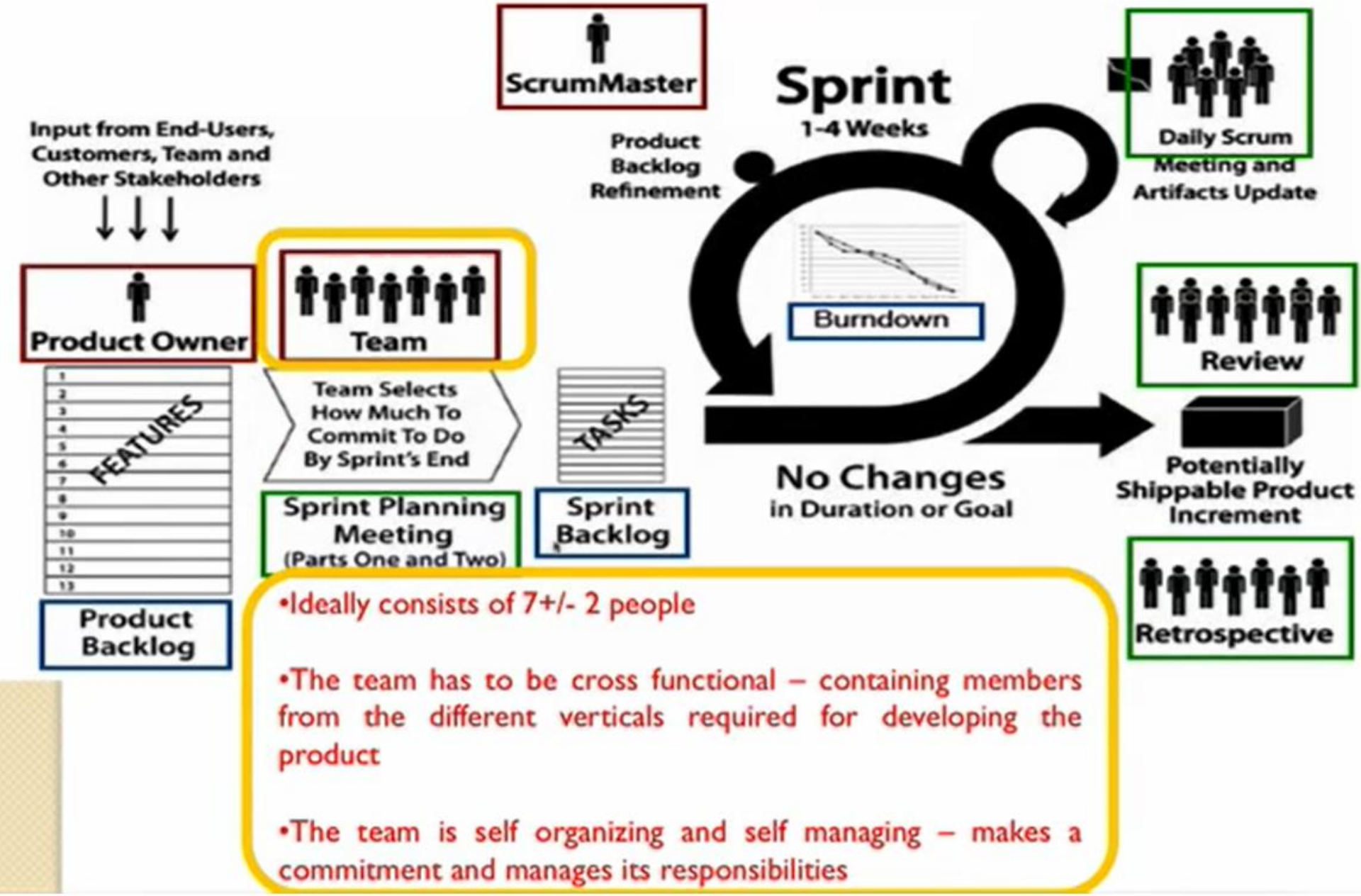
Scrum master:

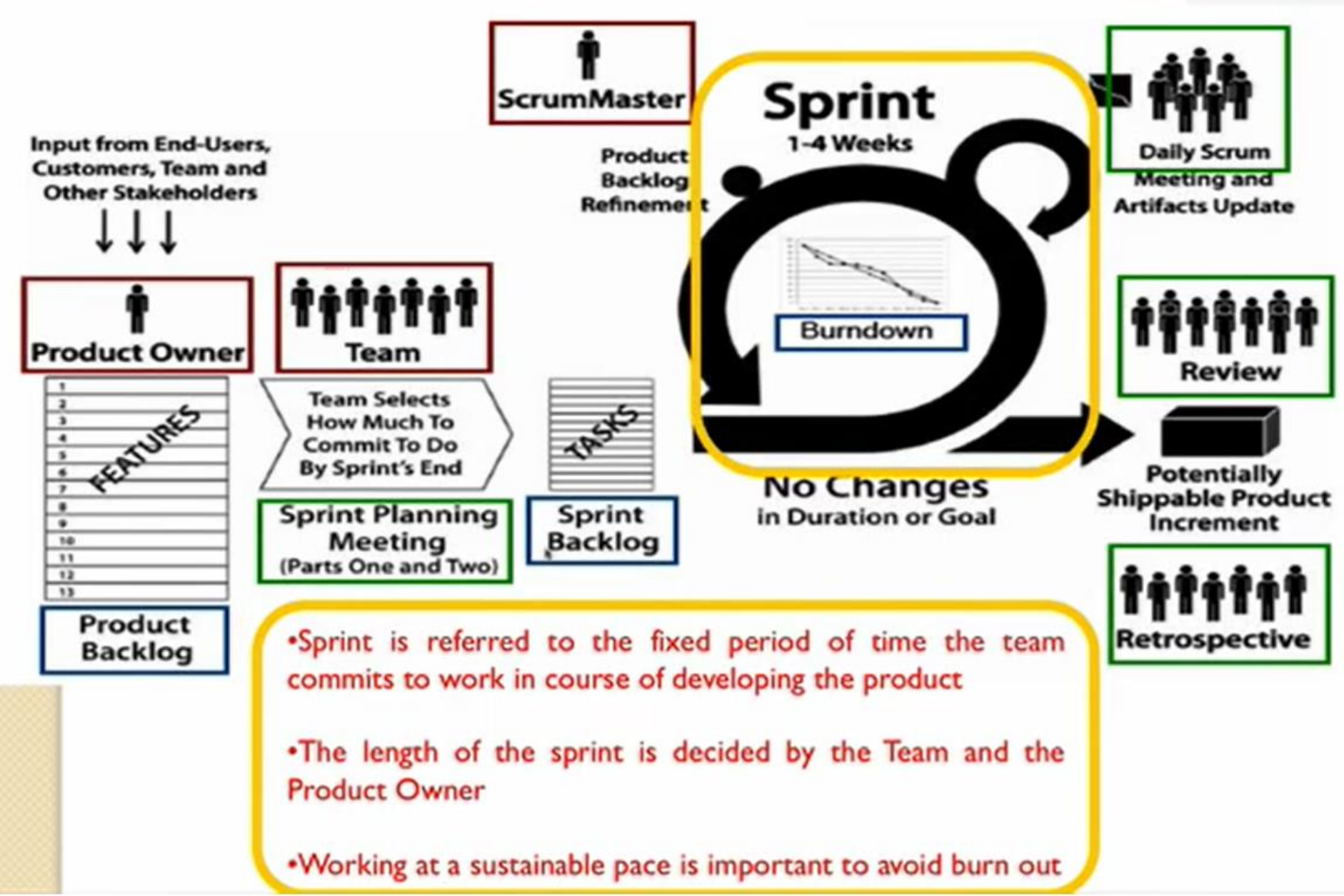
- Scrum master **serves the team** (helping them remove any and all impediments that surface),
- **protects the team** (from any outside disruption or interference) and
- **teaches & guides** the team's use of scrum
- He also train both team and product owner about **values and principles** of scrum.

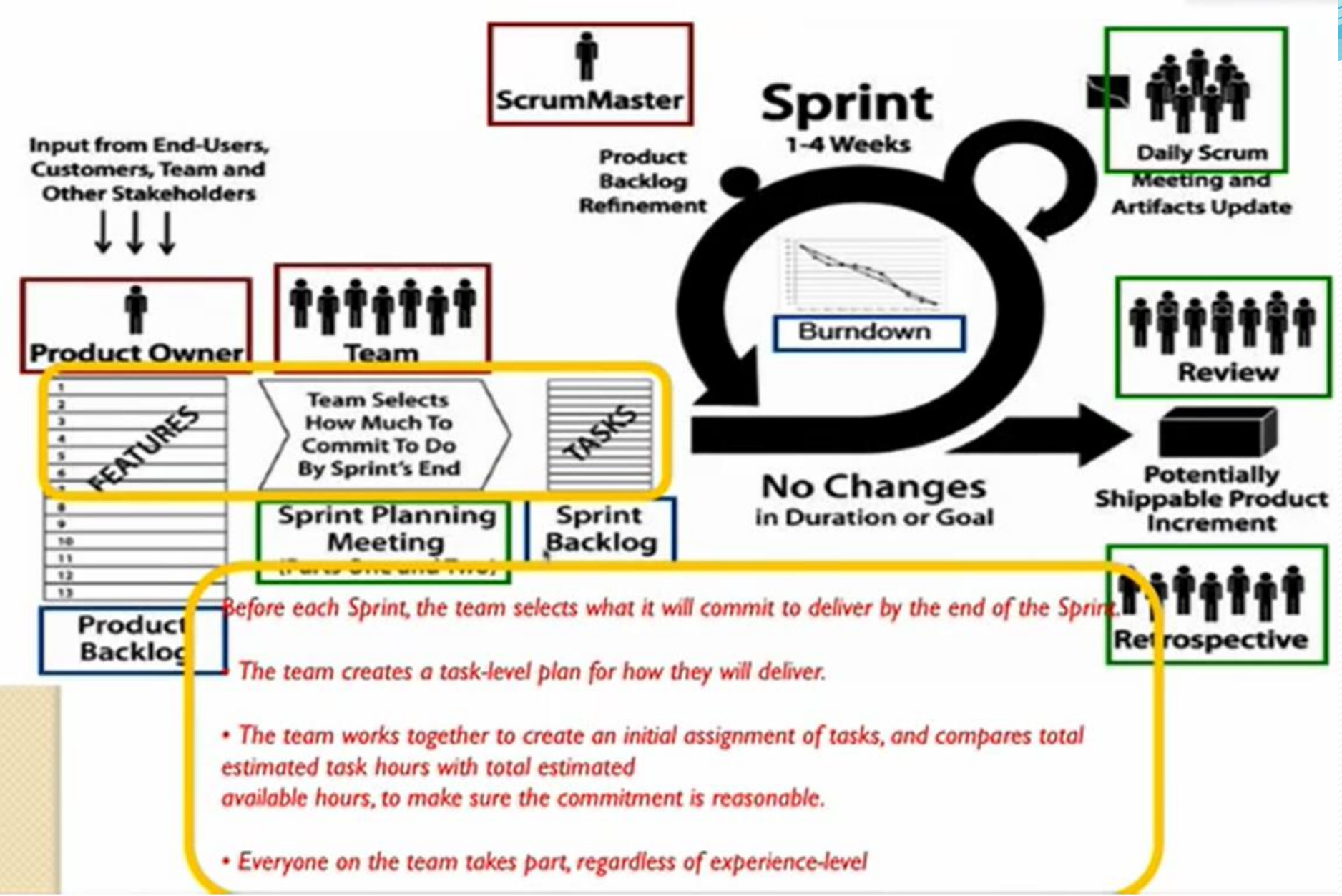
Scrum process in detail

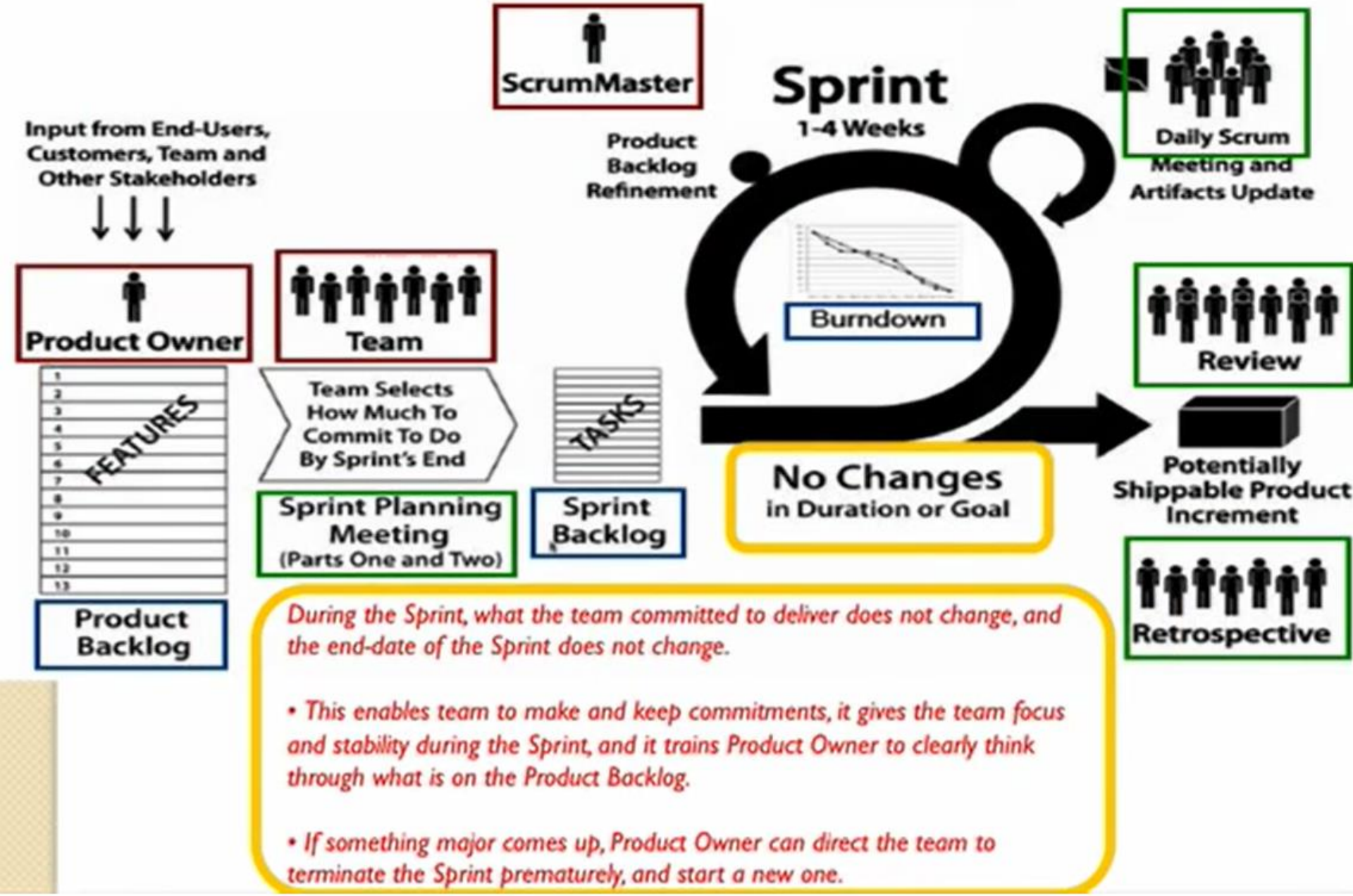

ScrumMaster
Sprint
1-4 Weeks
Input from End-Users, Customers, Team and Other Stakeholders
Product Backlog Refinement
Daily Scrum Meeting and Artifacts Update
Product Owner
Team
Burndown
Review
FEATURES
Team Selects How Much To Commit To Do By Sprint's End
TASKS
No Changes in Duration or Goal
Potentially Shippable Product Increment
Sprint Planning Meeting (Parts One and Two)
Sprint Backlog
Product Backlog
Retrospective

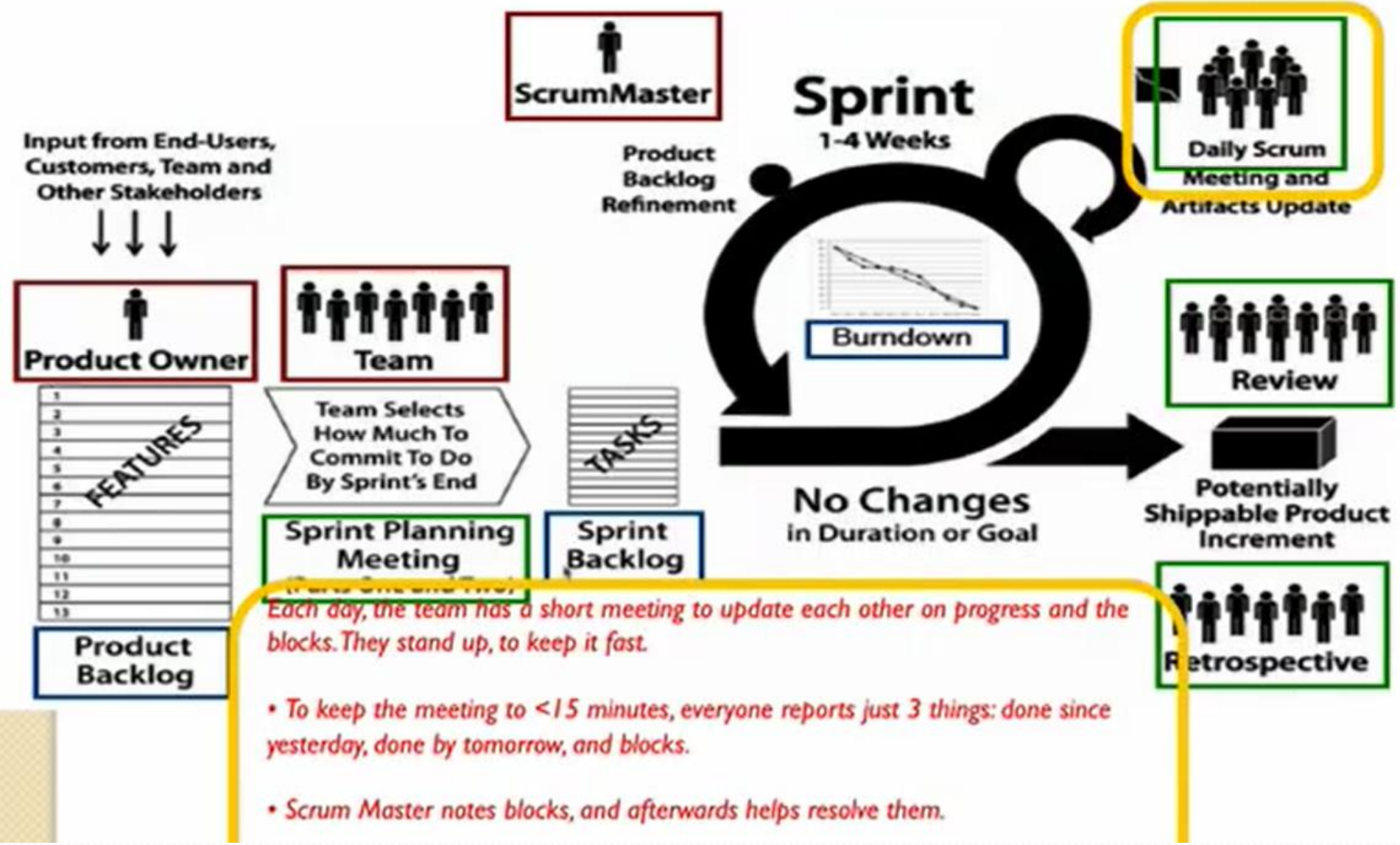

ScrumMaster
Sprint
1-4 Weeks
Input from End-Users, Customers, Team and Other Stakeholders
Product Backlog Refinement
Daily Scrum Meeting and Artifacts Update
Product Owner
Team
Burndown
Review
FEATURES
Team Selects How Much To Commit To Do By Sprint's End
TASKS
Sprint Planning Meeting (Parts One and Two)
Sprint Backlog
No Changes in Duration or Goal
Potentially Shippable Product Increment
Product Backlog
Retrospective
• The Product Backlog is the single master list of features, functionality etc ordered based on business value and risk
• The Product Backlog is constantly being revised by the Product Owner, to maximize the business success of the team's efforts.

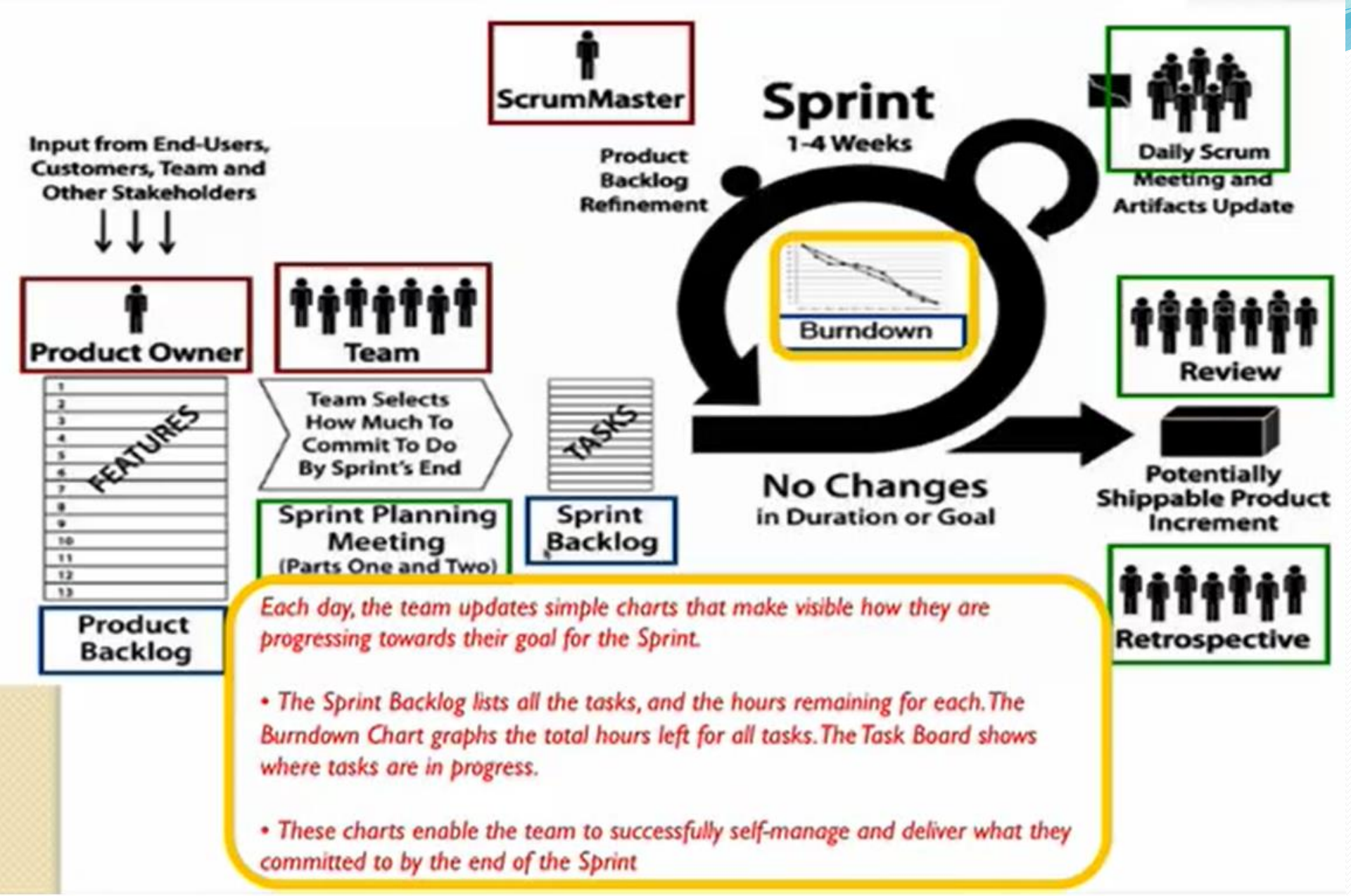

ScrumMaster
Sprint
1-4 Weeks
Input from End-Users, Customers, Team and Other Stakeholders
Product Backlog Refinement
Daily Scrum Meeting and Artifacts Update
Product Owner
Team
Burndown
Review
FEATURES
Team Selects How Much To Commit To Do By Sprint's End
TASKS
No Changes in Duration or Goal
Potentially Shippable Product Increment
Sprint Planning Meeting (Parts One and Two)
Sprint Backlog
Product Backlog
Retrospective
•Ideally consists of 7+/- 2 people
•The team has to be cross functional – containing members from the different verticals required for developing the product
•The team is self organizing and self managing – makes a commitment and manages its responsibilities

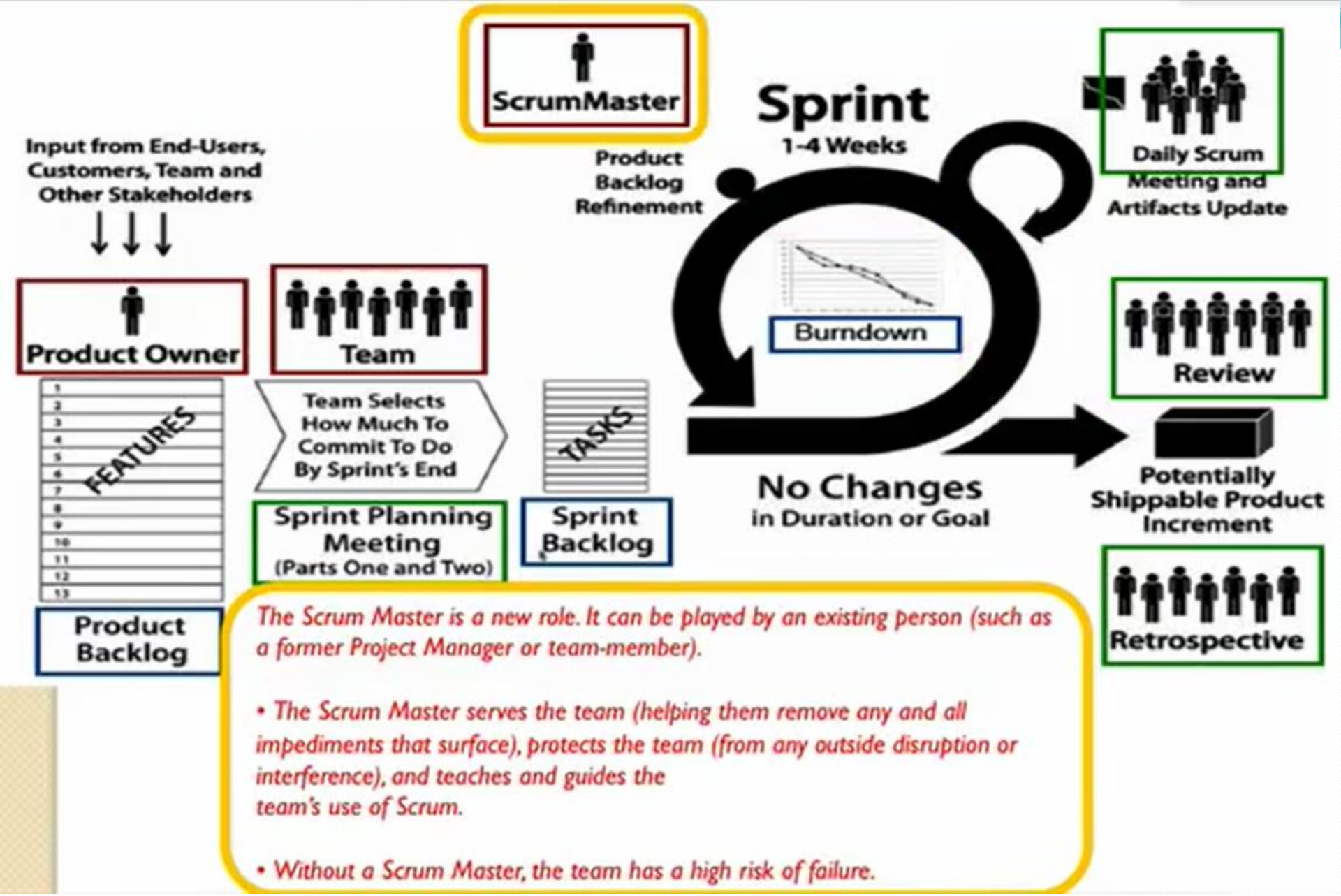

ScrumMaster
Sprint
1-4 Weeks
Daily Scrum
Meeting and
Artifacts Update
Input from End-Users,
Customers, Team and
Other Stakeholders
Product
Backlog
Refinement
Product Owner
Team
Burndown
Review
FEATURES
Team Selects
How Much To
Commit To Do
By Sprint's End
TASKS
Potentially
Shippable Product
Increment
Sprint Planning
Meeting
(Parts One and Two)
Sprint
Backlog
No Changes
in Duration or Goal
Product
Backlog
Retrospective
•Sprint is referred to the fixed period of time the team commits to work in course of developing the product
•The length of the sprint is decided by the Team and the Product Owner
•Working at a sustainable pace is important to avoid burn out

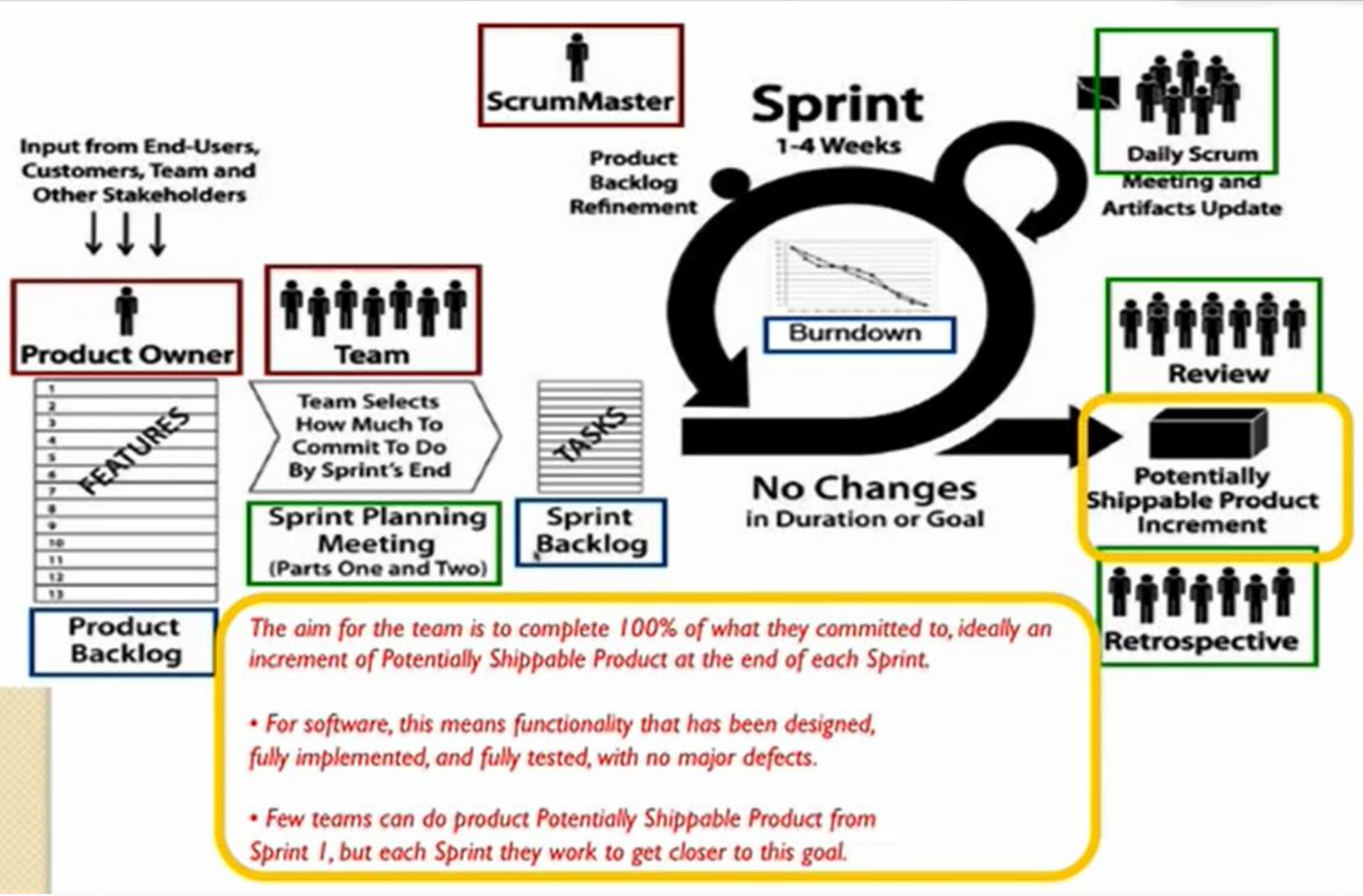

ScrumMaster
Sprint
1-4 Weeks
Input from End-Users, Customers, Team and Other Stakeholders
Product Backlog Refinement
Daily Scrum Meeting and Artifacts Update
Product Owner
Team
Burndown
Review
FEATURES
Team Selects How Much To Commit To Do By Sprint's End
TASKS
No Changes in Duration or Goal
Potentially Shippable Product Increment
Sprint Planning Meeting
Sprint Backlog
Product Backlog
Retrospective
Before each Sprint, the team selects what it will commit to deliver by the end of the Sprint.
• The team creates a task-level plan for how they will deliver.
• The team works together to create an initial assignment of tasks, and compares total estimated task hours with total estimated available hours, to make sure the commitment is reasonable.
• Everyone on the team takes part, regardless of experience-level

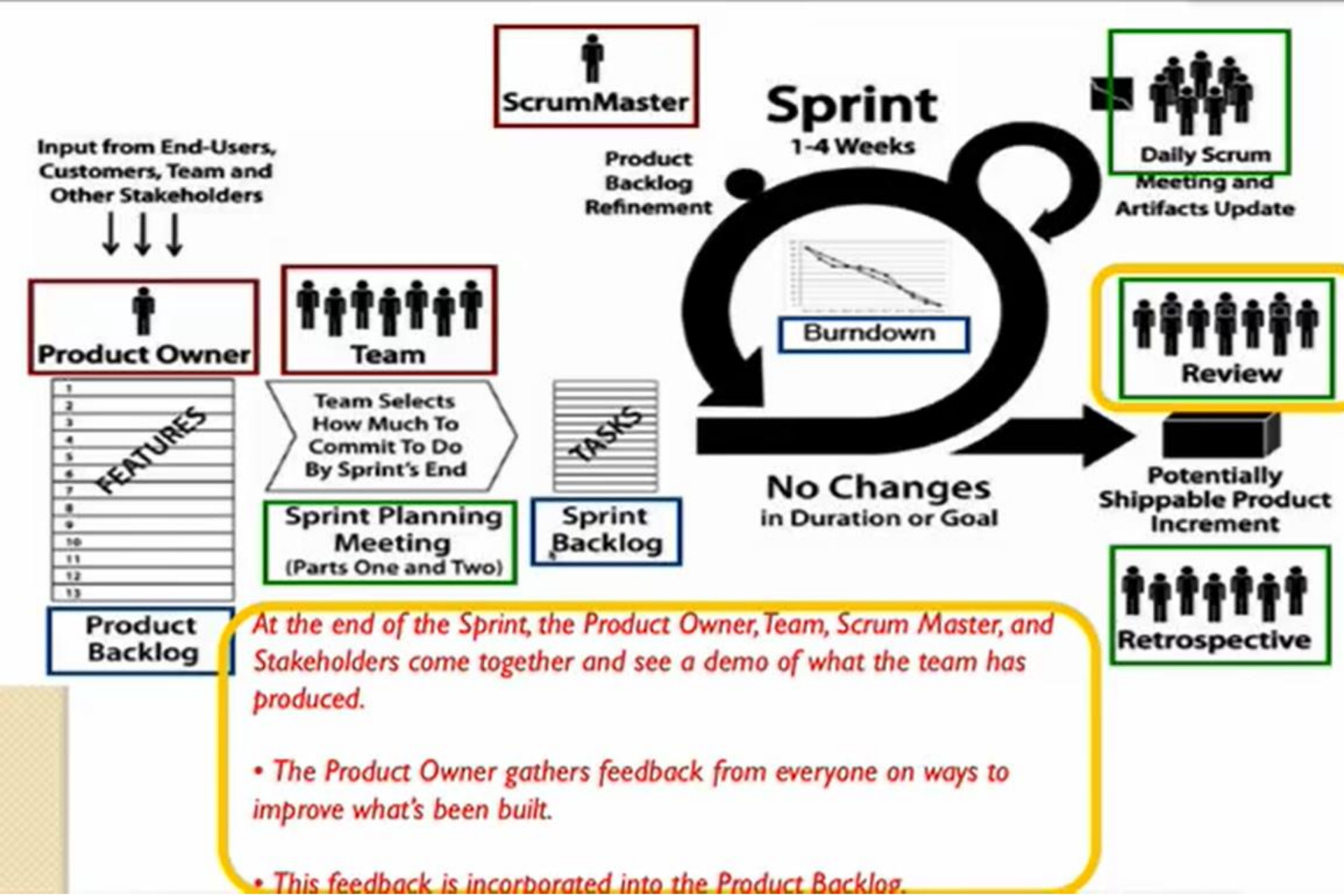

ScrumMaster
Sprint
1-4 Weeks
Input from End-Users, Customers, Team and Other Stakeholders
Product Backlog Refinement
Daily Scrum Meeting and Artifacts Update
Product Owner
Team
Burndown
Review
FEATURES
Team Selects How Much To Commit To Do By Sprint's End
TASKS
No Changes in Duration or Goal
Potentially Shippable Product Increment
Sprint Planning Meeting (Parts One and Two)
Sprint Backlog
Product Backlog
Retrospective
During the Sprint, what the team committed to deliver does not change, and the end-date of the Sprint does not change.
• This enables team to make and keep commitments, it gives the team focus and stability during the Sprint, and it trains Product Owner to clearly think through what is on the Product Backlog.
• If something major comes up, Product Owner can direct the team to terminate the Sprint prematurely, and start a new one.

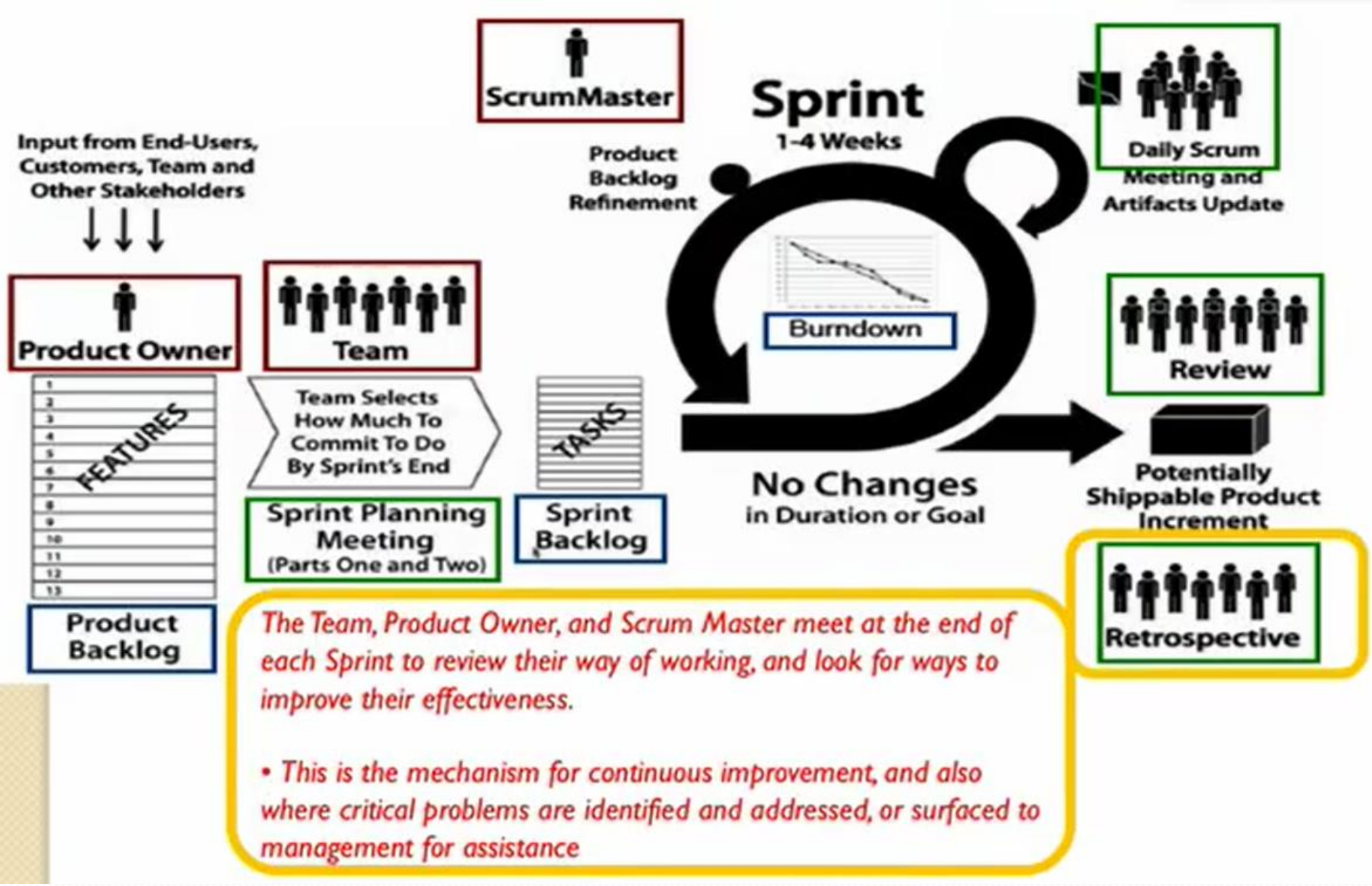

ScrumMaster
Sprint
1-4 Weeks
Input from End-Users, Customers, Team and Other Stakeholders
Product Backlog Refinement
Daily Scrum Meeting and Artifacts Update
Product Owner
Team
Burndown
Review
FEATURES
Team Selects How Much To Commit To Do By Sprint's End
TASKS
No Changes in Duration or Goal
Potentially Shippable Product Increment
Sprint Planning Meeting
Sprint Backlog
Product Backlog
Retrospective
Each day, the team has a short meeting to update each other on progress and the blocks. They stand up, to keep it fast.
• To keep the meeting to <15 minutes, everyone reports just 3 things: done since yesterday, done by tomorrow, and blocks.
• Scrum Master notes blocks, and afterwards helps resolve them.


ScrumMaster
Sprint
1-4 Weeks
Input from End-Users, Customers, Team and Other Stakeholders
Product Backlog Refinement
Daily Scrum Meeting and Artifacts Update
Product Owner
Team
Burndown
Review
FEATURES
Team Selects How Much To Commit To Do By Sprint's End
TASKS
No Changes in Duration or Goal
Potentially Shippable Product Increment
Sprint Planning Meeting (Parts One and Two)
Sprint Backlog
Retrospective
Product Backlog
Each day, the team updates simple charts that make visible how they are progressing towards their goal for the Sprint.
• The Sprint Backlog lists all the tasks, and the hours remaining for each. The Burndown Chart graphs the total hours left for all tasks. The Task Board shows where tasks are in progress.
• These charts enable the team to successfully self-manage and deliver what they committed to by the end of the Sprint


ScrumMaster
Sprint
1-4 Weeks
Input from End-Users, Customers, Team and Other Stakeholders
Product Backlog Refinement
Daily Scrum Meeting and Artifacts Update
Product Owner
Team
Burndown
Review
FEATURES
Team Selects How Much To Commit To Do By Sprint's End
TASKS
No Changes in Duration or Goal
Potentially Shippable Product Increment
Sprint Planning Meeting (Parts One and Two)
Sprint Backlog
Product Backlog
Retrospective
The Scrum Master is a new role. It can be played by an existing person (such as a former Project Manager or team-member).
• The Scrum Master serves the team (helping them remove any and all impediments that surface), protects the team (from any outside disruption or interference), and teaches and guides the team's use of Scrum.
• Without a Scrum Master, the team has a high risk of failure.


ScrumMaster
Sprint
1-4 Weeks
Input from End-Users, Customers, Team and Other Stakeholders
Product Backlog Refinement
Daily Scrum Meeting and Artifacts Update
Product Owner
Team
Burndown
Review
FEATURES
Team Selects How Much To Commit To Do By Sprint's End
TASKS
Potentially Shippable Product Increment
No Changes in Duration or Goal
Sprint Planning Meeting (Parts One and Two)
Sprint Backlog
Product Backlog
Retrospective
The aim for the team is to complete 100% of what they committed to, ideally an increment of Potentially Shippable Product at the end of each Sprint.
• For software, this means functionality that has been designed, fully implemented, and fully tested, with no major defects.
• Few teams can do product Potentially Shippable Product from Sprint 1, but each Sprint they work to get closer to this goal.


ScrumMaster
Sprint
1-4 Weeks
Input from End-Users, Customers, Team and Other Stakeholders
Product Backlog Refinement
Daily Scrum Meeting and Artifacts Update
Product Owner
Team
Burndown
Review
FEATURES
Team Selects How Much To Commit To Do By Sprint's End
TASKS
No Changes in Duration or Goal
Potentially Shippable Product Increment
Sprint Planning Meeting (Parts One and Two)
Sprint Backlog
Retrospective
Product Backlog
At the end of the Sprint, the Product Owner, Team, Scrum Master, and Stakeholders come together and see a demo of what the team has produced.
• The Product Owner gathers feedback from everyone on ways to improve what's been built.
• This feedback is incorporated into the Product Backlog.


ScrumMaster
Sprint
1-4 Weeks
Input from End-Users, Customers, Team and Other Stakeholders
Product Backlog Refinement
Daily Scrum Meeting and Artifacts Update
Product Owner
Team
Burndown
Review
FEATURES
Team Selects How Much To Commit To Do By Sprint's End
TASKS
Sprint Planning Meeting (Parts One and Two)
Sprint Backlog
No Changes in Duration or Goal
Potentially Shippable Product Increment
Product Backlog
Retrospective
The Team, Product Owner, and Scrum Master meet at the end of each Sprint to review their way of working, and look for ways to improve their effectiveness.
• This is the mechanism for continuous improvement, and also where critical problems are identified and addressed, or surfaced to management for assistance

Scrum

- Scrum—distinguishing features
  - Development work is partitioned into "**packets**".
  - **Testing and documentation are on-going** as the product is constructed.
  - Work occurs in "**sprints**" and is derived from a "**backlog**" of existing requirements.
  - **Meetings are very short** and sometimes conducted without chairs. A team leader, called a ***Scrum master*****,** leads the meeting and assesses the responses from each person.
  - "**demos**" are delivered to the customer with the time-box allocated

Scrum

- Scrum incorporates a set of process patterns that emphasize
  - **project priorities,**
  - **compartmentalized work units,**
  - **communication, and**
  - **frequent customer feedback.**

Scrum

- ***Backlog***—*A prioritized list of project requirements*,

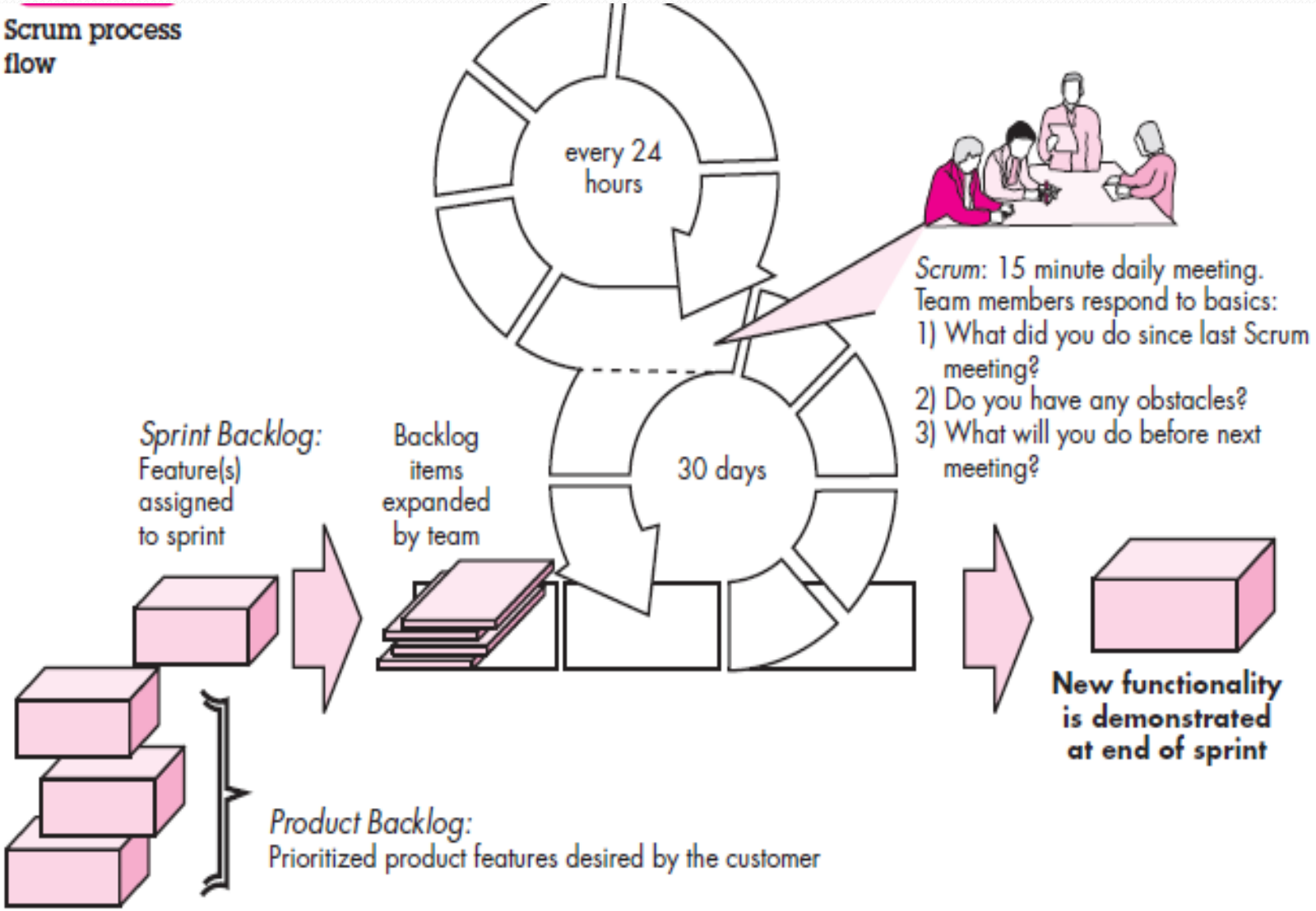
  The product manager assesses the backlog and updates priorities as required.
- **Sprints**—consist of **work units** that are required to achieve a requirement defined in the backlog that must be fit into a **predefined time-box(typically 30 days).**
- Changes (e.g., backlog work items) are not introduced during the sprint. Hence, the sprint allows team members to work in a short-term, but stable environment.

- **Daily Scrum Meeting**: The Daily Scrum is the key inspect and adapt meeting during a Sprint. During the Sprint execution , the scrum Team meets every day, for Daily Scrum meeting and inspects the progress and ensures communication flow inside the Team. **It is a short (15 minutes) meeting**.
- Time box: 15 Minutes
- Attendees: Scrum Master, Product Owner(Optional) and Scrum Team
- It is held at the same time at same place each day. During the meeting, each Team member explains:
  - What have I accomplished since the last meeting?
  - What am I going to do before the next meeting?
  - What obstacles are in my way?

Sprint backlog:

- The Sprint Backlog is the set of Product Backlog items selected for the Sprint, plus a plan for delivering the product Increment and realizing the Sprint Goal.
- The Sprint Backlog is a forecast by the Team about what functionality will be made available in the next Increment and the work needed to deliver that functionality as a working product Increment. The Sprint Backlog is a plan with enough detail that can be understood but the Team to track in the Daily Scrum. The Team modifies the Sprint Backlog throughout the Sprint, and the Sprint
- Backlog emerges during the Sprint. This emergence occurs as the Team works through the plan and learns more about the work needed to achieve the Sprint Goal.

Scrum--The overall flow of the Scrum process



Summary

- It is **an iterative, incremental** framework
- Sprints – cycles of work developed, duration: 2 - 4 weeks; occur one after another **without pause**
- **Time boxed** – they end whether or not the work ends
- At the beginning, cross-functional team forms the priority list based on **customer requirements**
- During the sprint the chosen items **do not change**
- Everyday inspection and adjustment
- End of the sprint, review with stakeholders
- **Feedbacks** are taken and incorporated into the sprint
- End of sprint, **fully tested product** is formed as per customer requirements

Other Agile Process models

Many agile process models have been proposed and are in use **across the industry**. Among them **most common** are:

- Extreme Programming ("**XP**")
- Adaptive Software Development (**ASD**)
- **Scrum**
- **Crystal**
- Feature Drive Development (**FDD**)
- Lean Software Development (**LSD**)
- Agile Modeling (**AM**)
- Agile Unified Process (**AUP**)

- END OF AGILE DEVELOPMENT

Requirements Engineering