CSE 470
Software Engineering
Agile Methodology

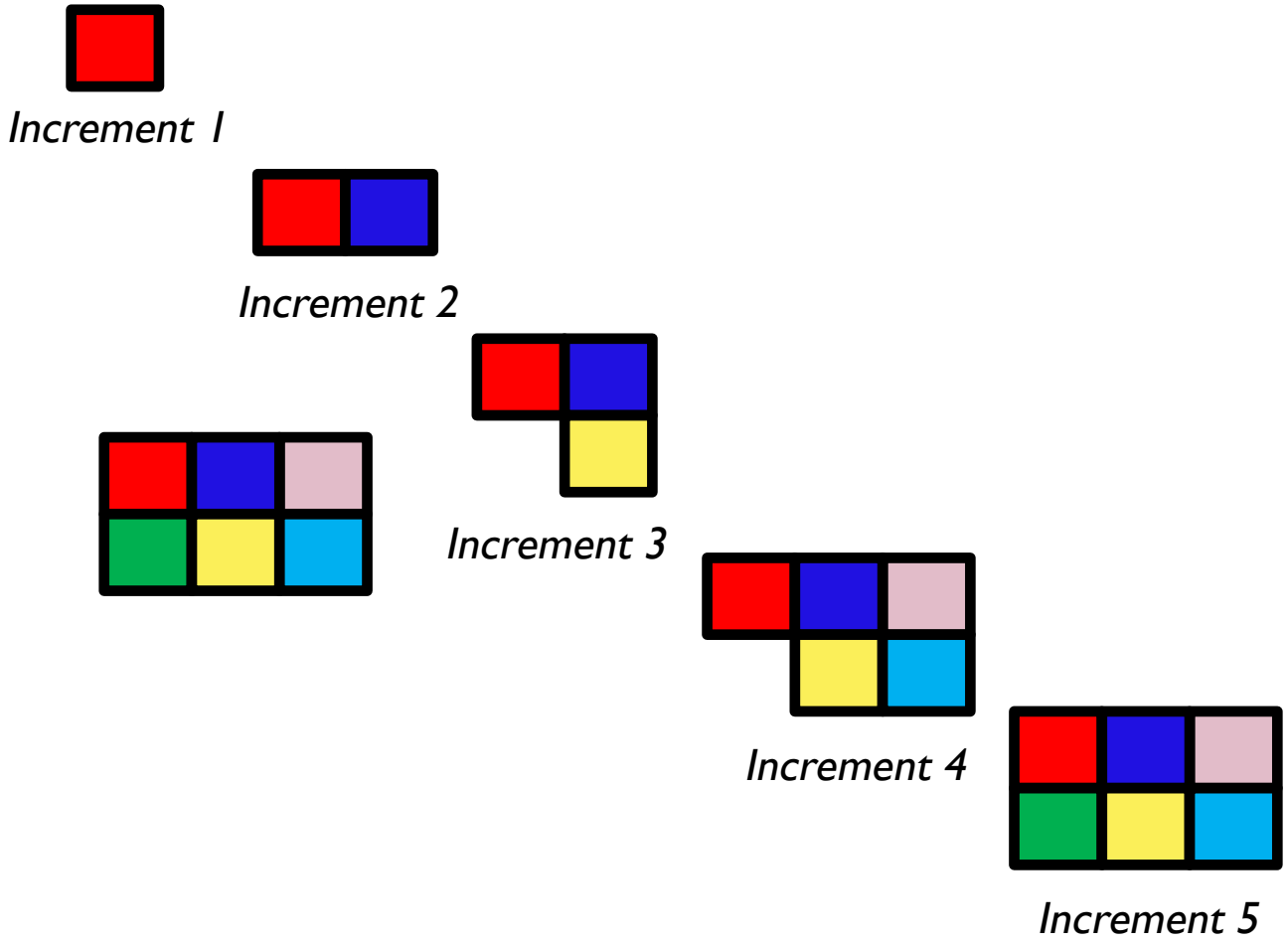
Imran Zahid
Lecturer
Computer Science and Engineering, BRAC University

Recap

Incremental Process Model

- For example, in first increment Login module is delivered.
- In second increment, another new module such as Navigation module is also added.
- In third, one more added.

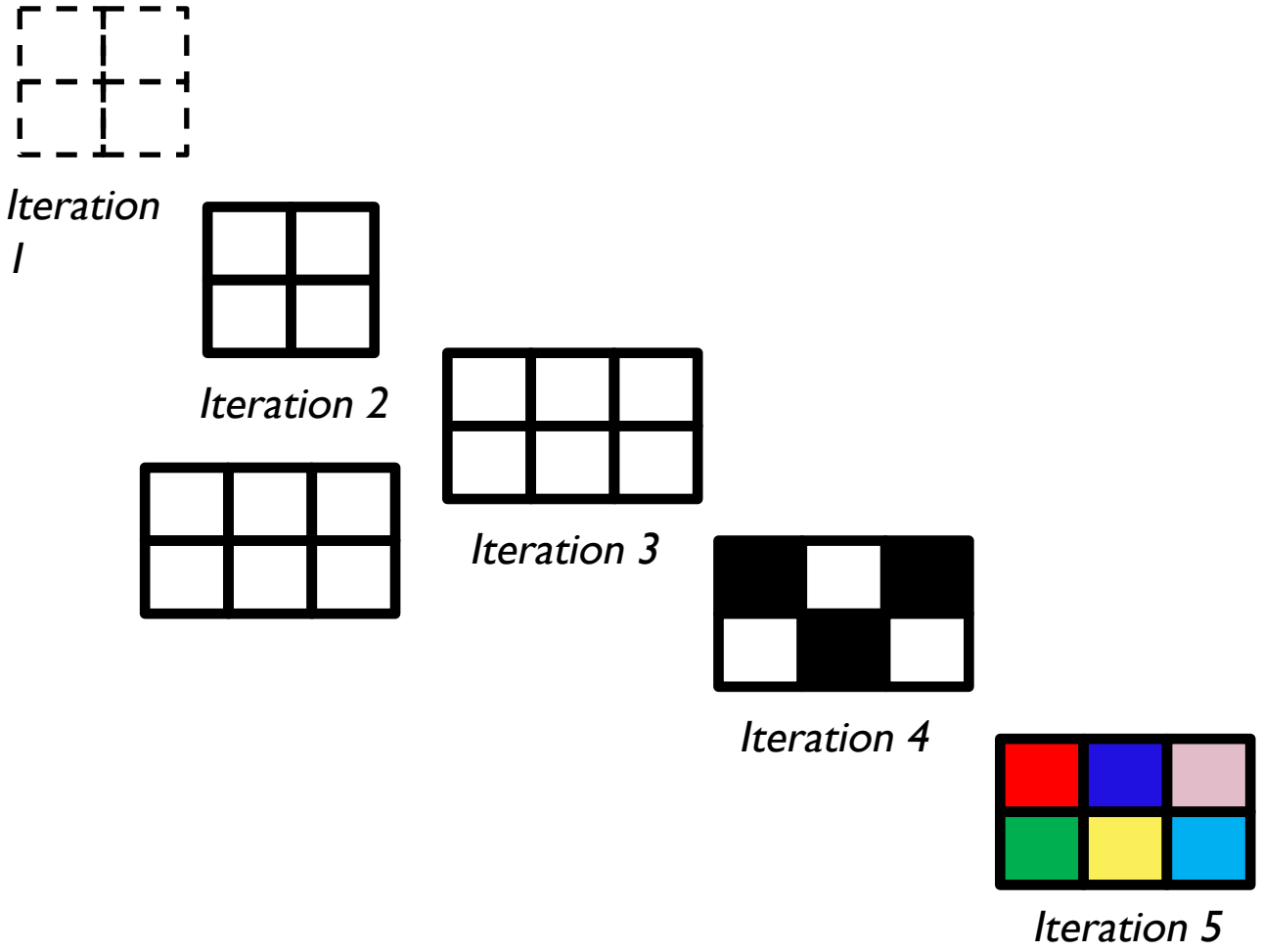
That is, this model “adds onto” new section as increments.



Iterative Process Model

- For example, in first iteration Login module is delivered.
- In second iteration, Login module is updated again.
- In third iteration, Navigation module is added . And in fourth iteration some refinement is done.

That is, this model “changes or reworks” on same section in iterations until customer accepts it.

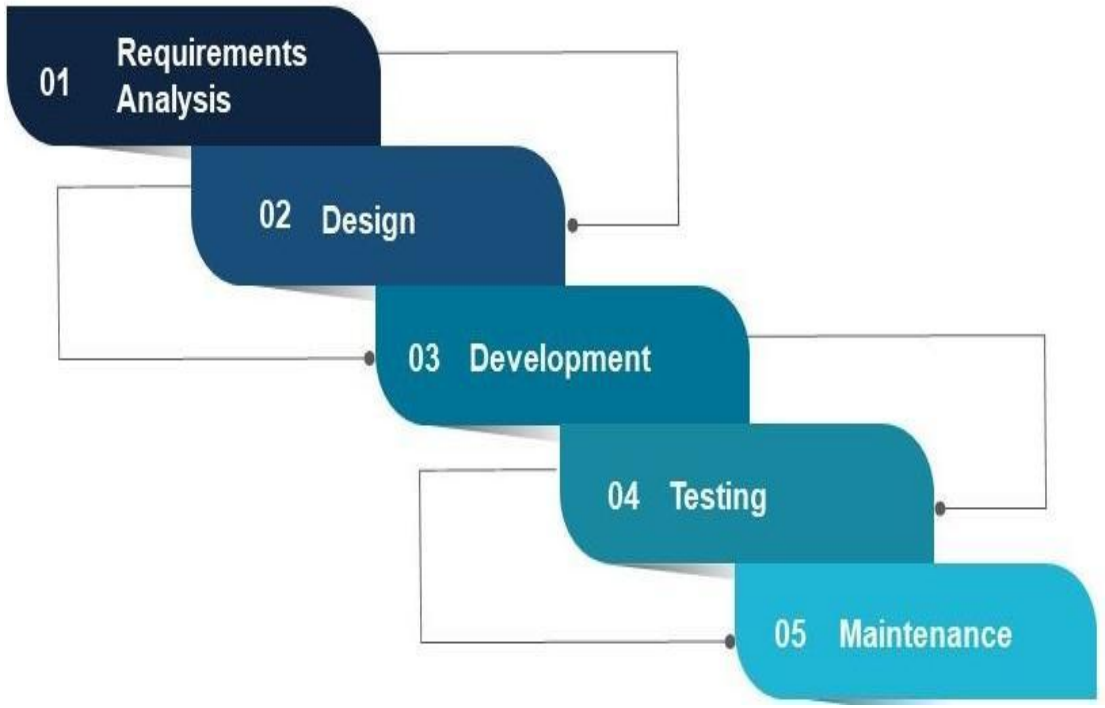


Agile Methodology

Why we need Agile

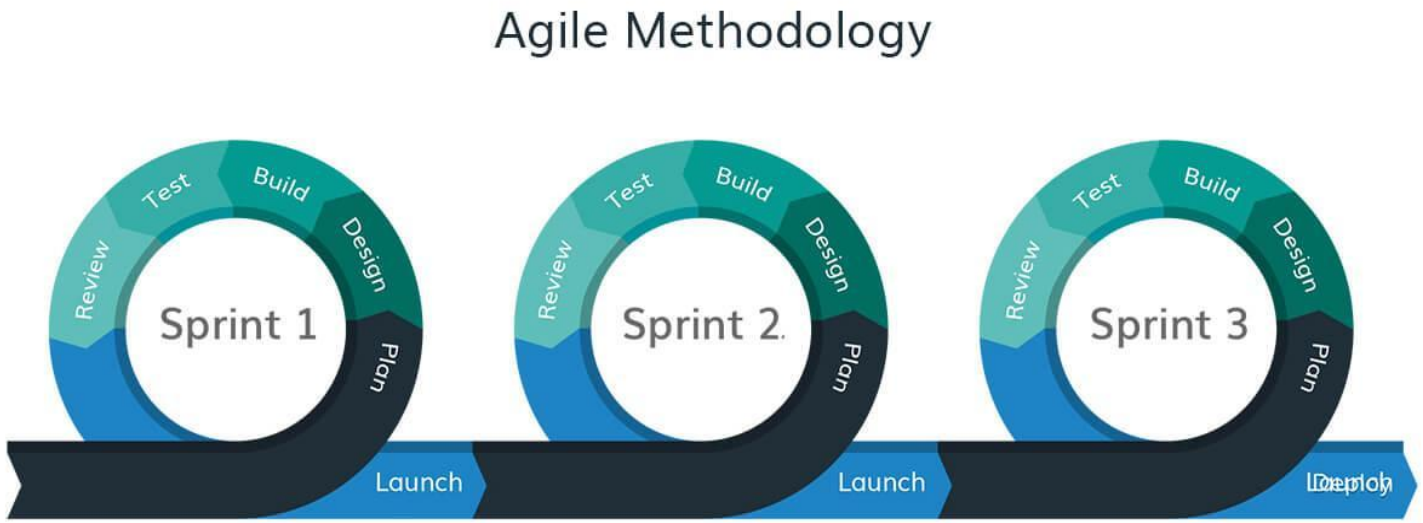
1. Monolithic Software: The ready product comes at the end of the process. Nowadays, the requirement of software changes frequently, it can even change several times a day.

2. For live systems like Amazon and Facebook, a few seconds of downtime can cause a lot of problems. Changes in such cases need to be smooth enough so that customers' interactions are not interrupted.



What is ‘Agile?’

- Agile is a set of principles and values. It is a practice to be followed.
- Agile is a combination of Iterative and Incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software.



What is ‘Agile?’

Uncovering better ways of developing software by doing it and helping others do it.

Individuals and interactions	*over*	***processes and tools***
Working software	*over*	***comprehensive documentation***
Customer collaboration	*over*	***contract negotiation***
Responding to change	*over*	***following a plan***

Agile Methodologies

- Extreme Programming
- Agile Unified Process
- Scrum

Agile - Extreme Programming

Extreme Programming

- Extreme Programming (XP) is an agile software development framework that aims to produce higher quality software and higher quality of life for the development team.
- XP is the most specific of the agile frameworks regarding appropriate engineering practices for software development.

Extreme Programming - When to Use?

- Dynamically changing software requirements
- Risks caused by fixed-time projects using new technology
- Small, co-located extended development team
- The technology you are using allows for automated unit and functional tests

XP Principles or Practices

Incremental planning	Refactoring	
Small releases	Pair programming	Sustainable pace
Simple design	Collective ownership	On-site customer
Test-first development	Continuous integration	

Weekly Cycle

- The Weekly Cycle is synonymous with an iteration.
- In the case of XP, the team meets on the first day of the week to reflect on progress to date, the customer picks the stories they would like delivered in that week, and the team determines how they will approach those stories.
- The goal by the end of the week is to have running tested features that realize the selected stories.

Quarterly Cycle

- The Quarterly Cycle is synonymous with a release. The purpose is to keep the detailed work of each weekly cycle in the context of the overall project. The customer lays out the overall plan for the team in terms of features desired within a particular quarter.
- Remember when planning a quarterly cycle the information about any particular story is at a relatively high level, the order of story delivery within a Quarterly Cycle can change and the stories included in the Quarterly Cycle may change.

Stories

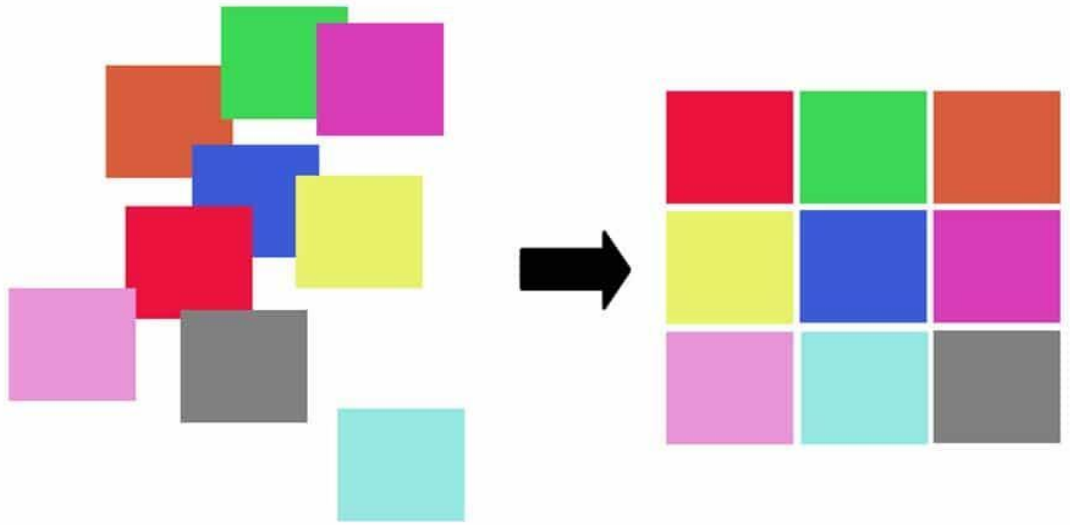
- Describe what the product should do in terms meaningful to customers and users.
- These stories are intended to be short descriptions of things users want to be able to do with the product that can be used for planning and serve as reminders for more detailed conversations when the team gets around to realizing that particular story.

Test First Development

- Writes unit test for each method to identify problems and resolve it
- Test First Development (TDD) is used to ensure better code quality and less error.
- Instead of following the normal path of:
  - develop code -> write tests -> run tests
- The practice of Test-First Development follows the path of
  - Write failing automated test -> Run failing test -> develop code to make test pass -> run test -> repeat

Refactoring

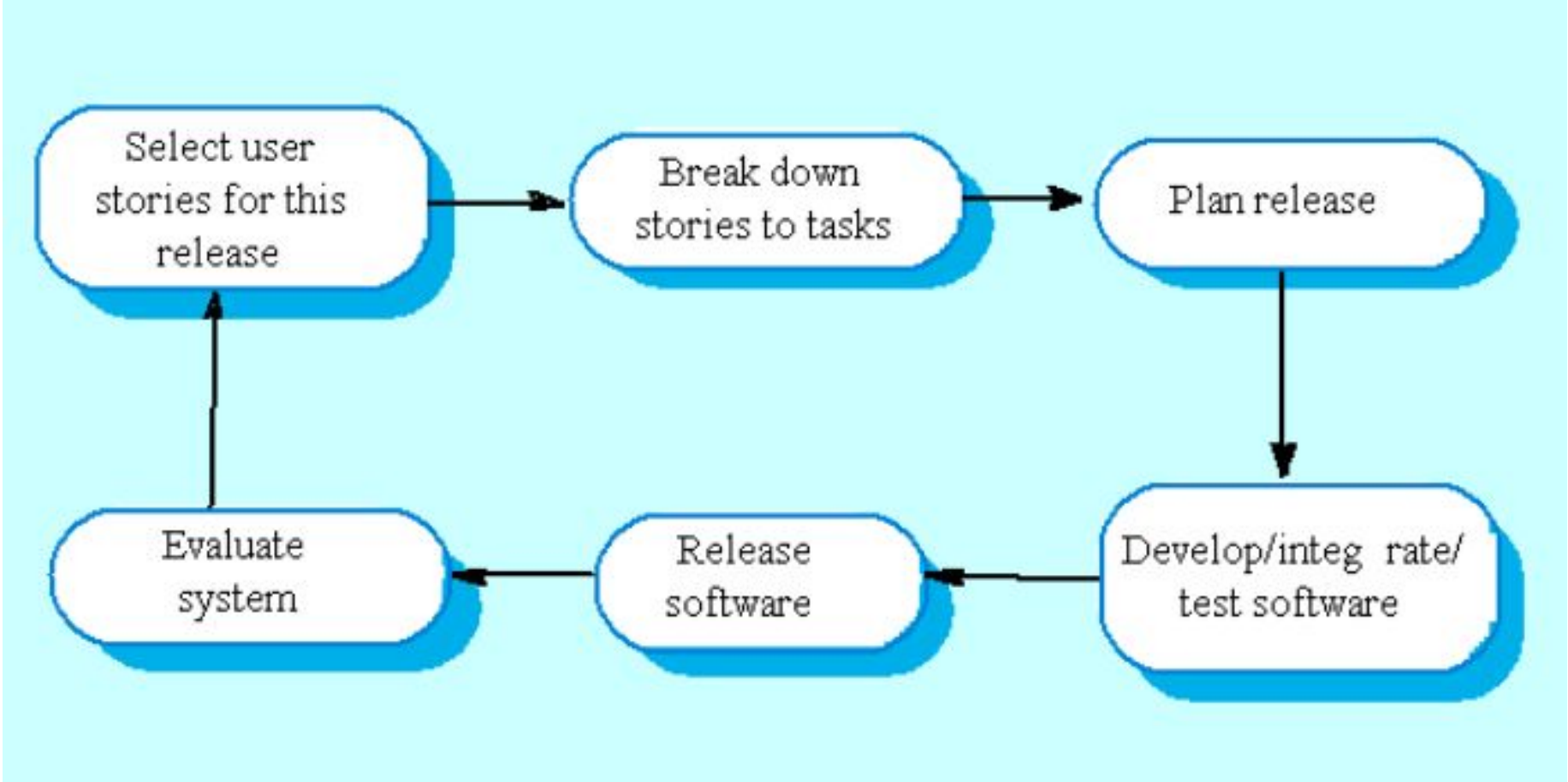
- It’s a technique for restructuring existing code by changing its internal structure without changing its external behaviour.
- Refactoring includes reducing duplicate codes, breaking long classes/methods into small ones, appropriate variable naming and many more.
- It is done for requirement change, improving design and easy extensibility of software.



Pair Programming

- Pair Programming means all production software is developed by two people sitting at the same machine. You effectively get a continuous code review and quicker response to nagging problems that may stop one person dead in their tracks.
- Two people code in share, they switch roles from observer to navigator frequently
- Observing code produces better code, so less business cost. Does not actually take twice as long because they are able to work through problems quickly and they stay more focused on the task at hand, thereby creating less code to accomplish the same thing.

XP Workflow


Select user stories for this release
Break down stories to tasks
Plan release
Develop/integ rate/ test software
Release software
Evaluate system

Agile - Agile Unified Process

Agile Unified Process

- Another Agile Software Development Methodology
- Agile Unified Process (AUP) is a simplified version of the Rational Unified Process (RUP).

AUP Phases

Inception

- Try to identify the business scope of the project, initial requirements and potential solution of the problem.

Elaboration

- Design and prove the solution architecture, more requirements can be extracted

Construction

- Continuous implementation and testing (specially unit testing) in form of iterations

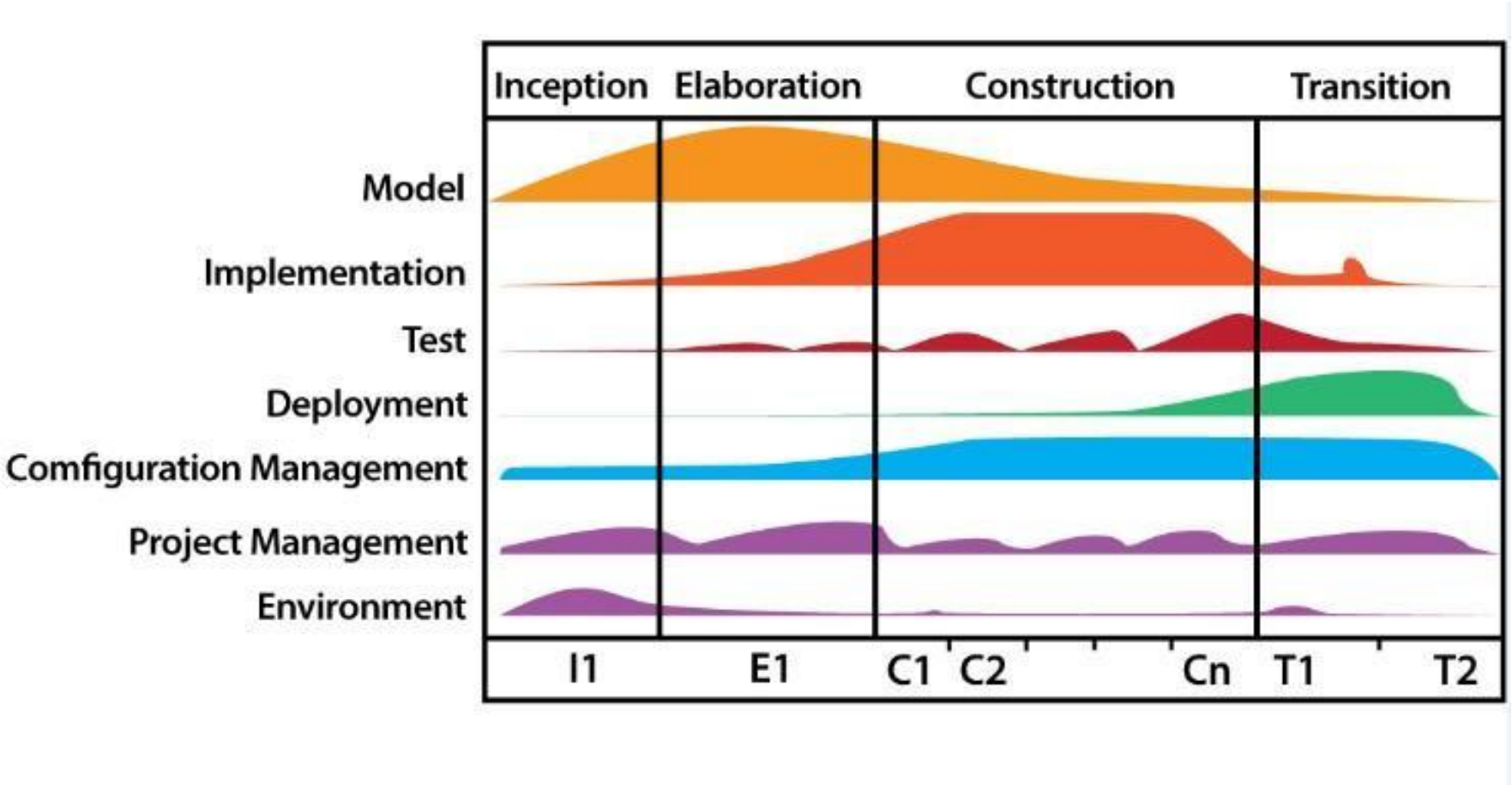
Transition

- Lastly, deploy the system in production environment and adjust feedbacks.

AUP Disciplines

- **Model**
- **Implementation**
- **Test**
- **Deployment**
- **Configuration Management:** Manage access to project artifacts/versions. This includes not only tracking artifact versions over time but also controlling and managing changes to them.
- **Project Management:** Direct the activities that take place within the project. This includes managing risks, directing people (assigning tasks, tracking progress, etc.), and coordinating with people and systems outside the scope of the project to be sure that it is delivered on time and within budget.
- **Environment:** Support the rest of the effort by ensuring that the proper process, guidance (standards and guidelines), and tools (hardware, software, etc.) are available for the team as needed.

AUP Disciplines



AUP - When to Use?

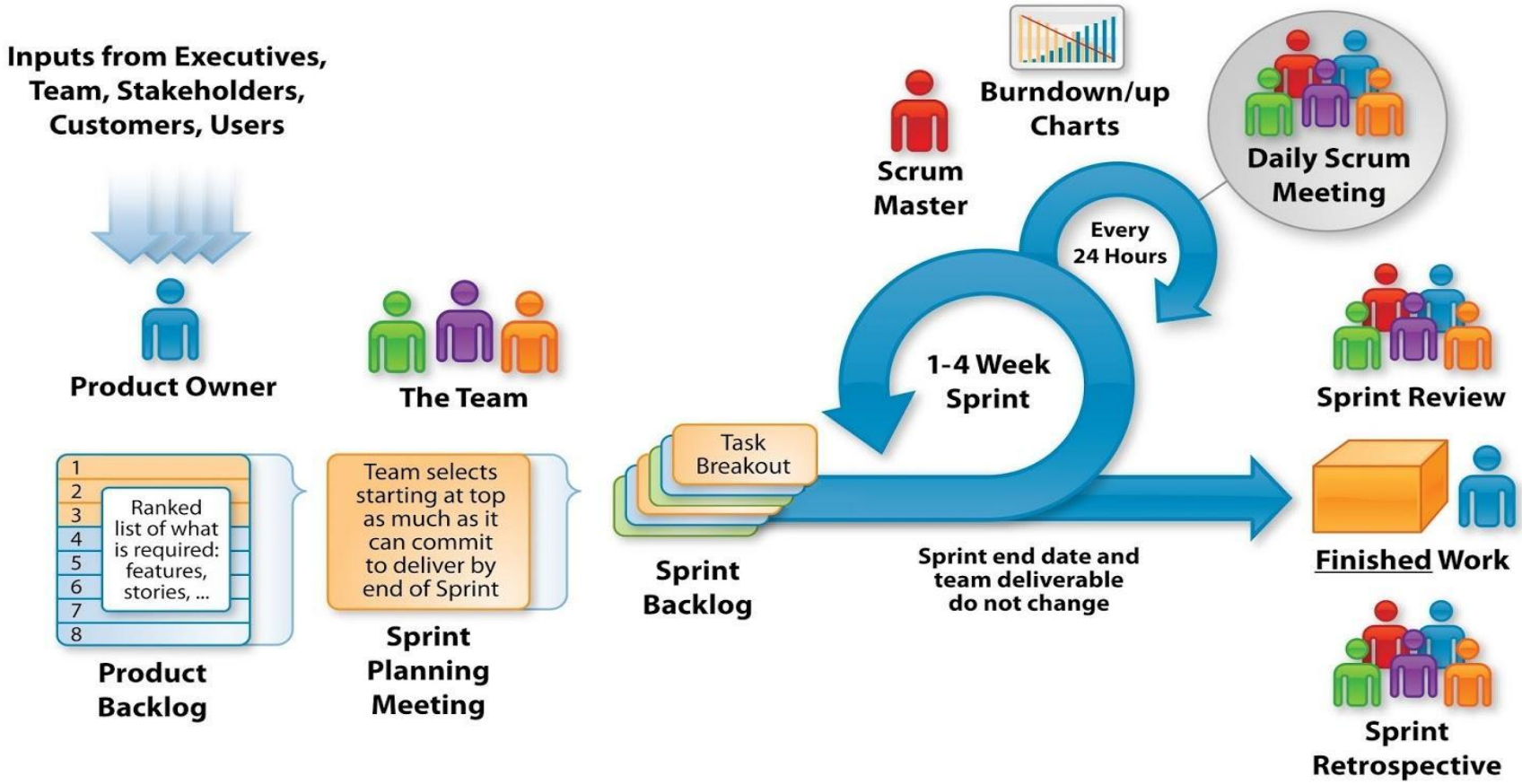
- The project needs a more structured agile approach with phases (like Inception, Elaboration, Construction, Transition).
- Documentation, traceability, and a phased approach are required, but a fully traditional RUP is too rigid.
- There is a need for stakeholder feedback while adhering to a predefined lifecycle with minimal overhead.

Agile - Scrum

Scrum

- It is the most widely used Agile S/W development method for project management
- The full software is delivered in 7-30 days iterations

The Agile - Scrum Framework



Scrum Framework

Roles

- Product Owner
- Scrum Master
- Team

Ceremonies

- Sprint planning
- Sprint review and Sprint retrospective
- Daily scrum meeting

Artifacts

- Product backlog
- Sprint backlog
- Burndown charts

Scrum Roles


YOU
STAKEHOLDERS
SCRUM MASTER
PRODUCT OWNER
SCRUM TEAM
DEVELOPMENT TEAM

Scrum Roles

Product Owner

- Possibly a Product Manager or Project Sponsor
- Decides features, release date, prioritization, $$$

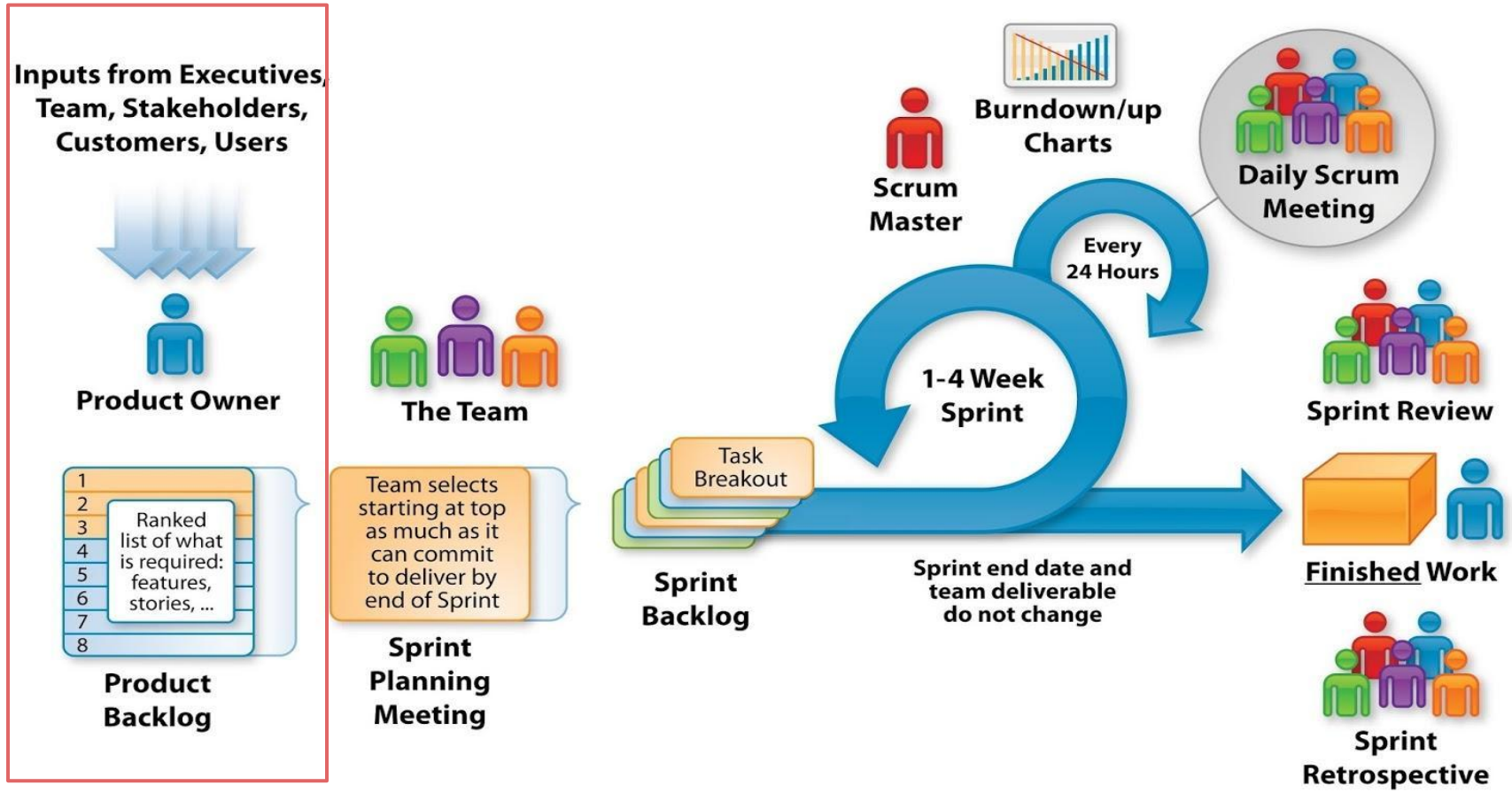
Scrum Master

- Typically a Project Manager or Team Leader
- Responsible for enacting Scrum values and practices
- Remove impediments / politics, keeps everyone productive

Project Team

- 5-10 members; Teams are self-organizing
- Cross-functional: QA, Programmers, UI Designers, etc.
- Membership should change only between sprints

The Agile - Scrum Framework



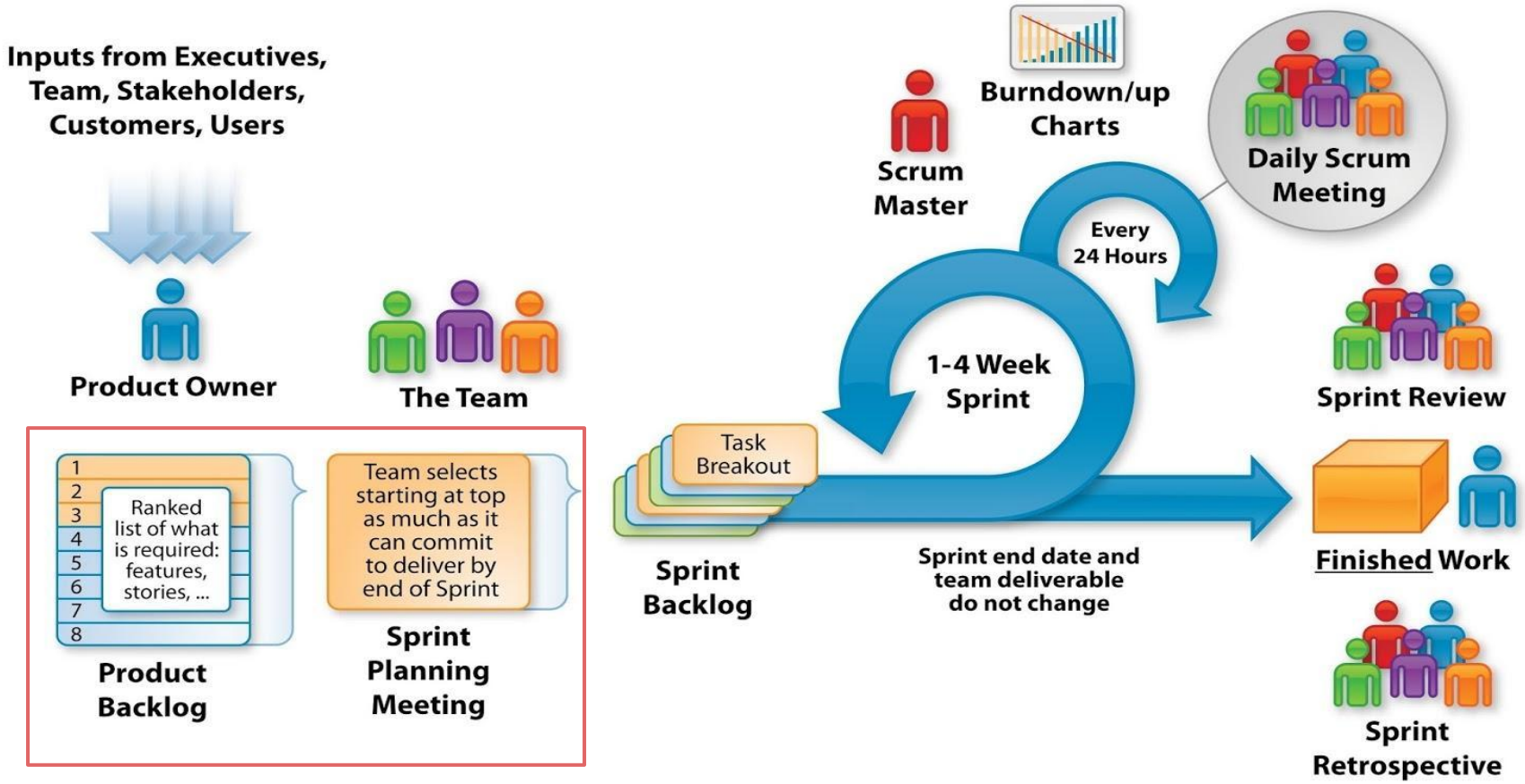
Artifact - Product Backlog

- The requirements
- A list of all desired work on project
- Ideally expressed as a list of user stories along with "story points", such that each item has value to users or customers of the product
- Prioritized by the product owner
- Reprioritized at start of each sprint

Artifact - Product Backlog

Feature	Backlog item	Estimate
A	Allow a guest to make a reservation	3 (story points)
B	As a guest, I want to cancel a reservation.	5
C	As a guest, I want to change the dates of a reservation.	3
D	As a hotel employee, I can run RevPAR reports (revenue-per-available-room)	8
E	Improve exception handling	8
…	…	30
…	…	50

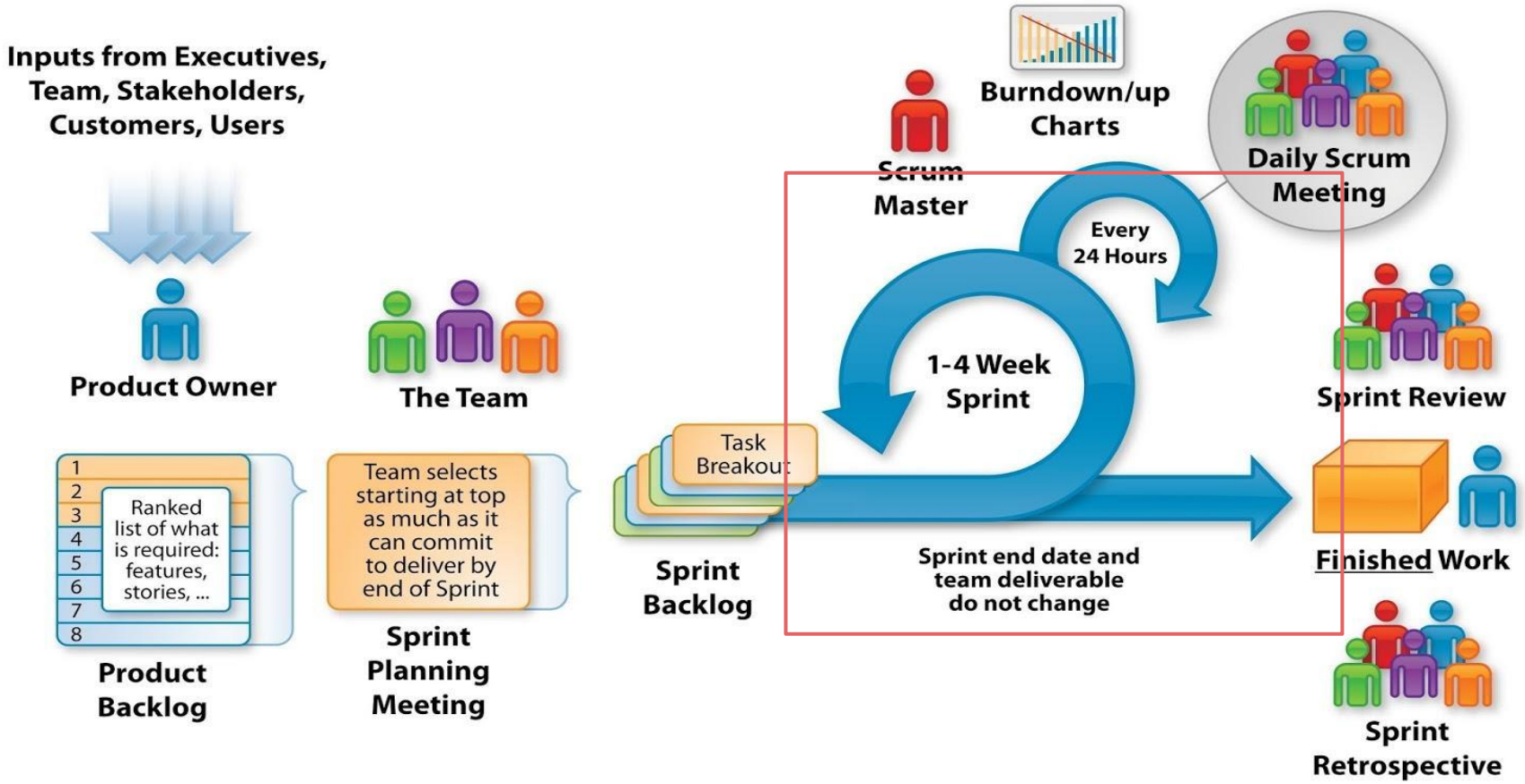
The Agile - Scrum Framework



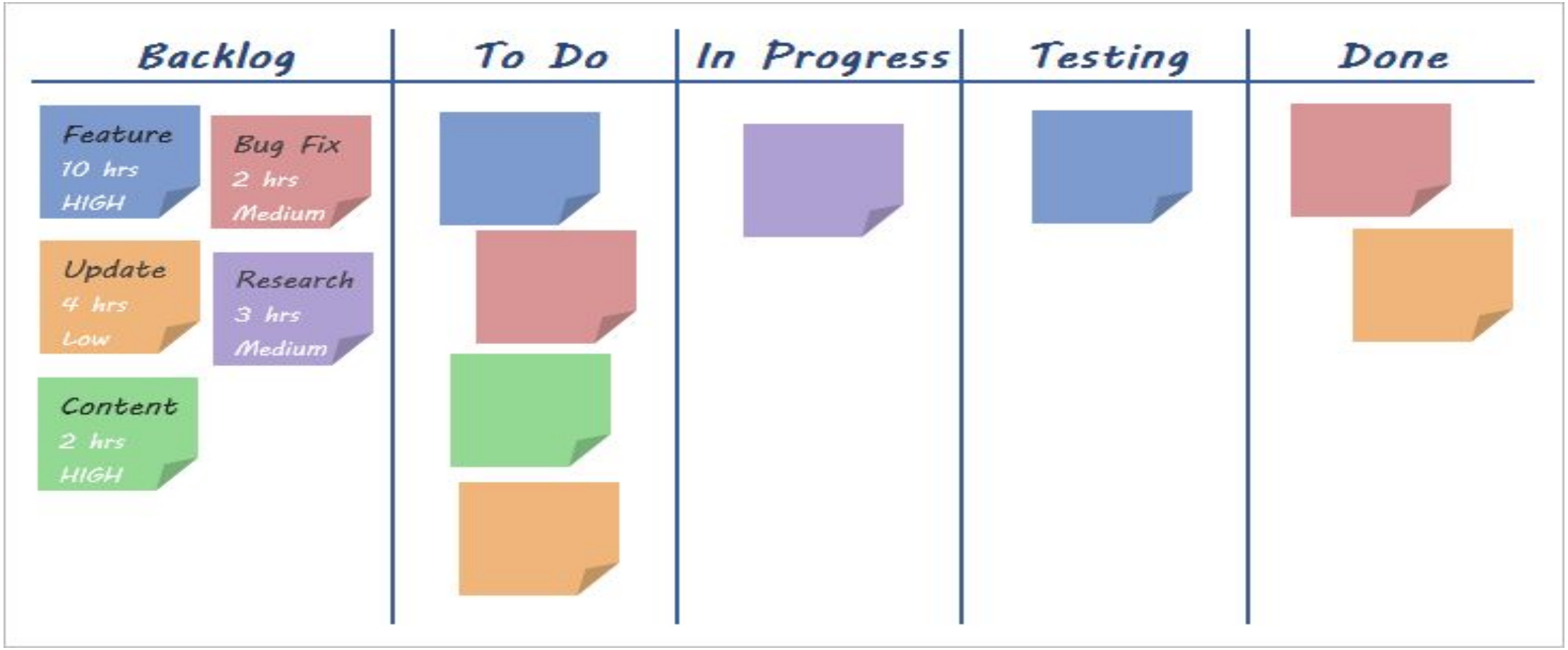
Artifact - Sprint Backlog

- High priority features are selected based on points from product backlog.
- These features will be covered within that 7-30 days time slot called as sprint.
- Estimated work remaining is updated daily
- Any team member can add, delete or change the sprint backlog

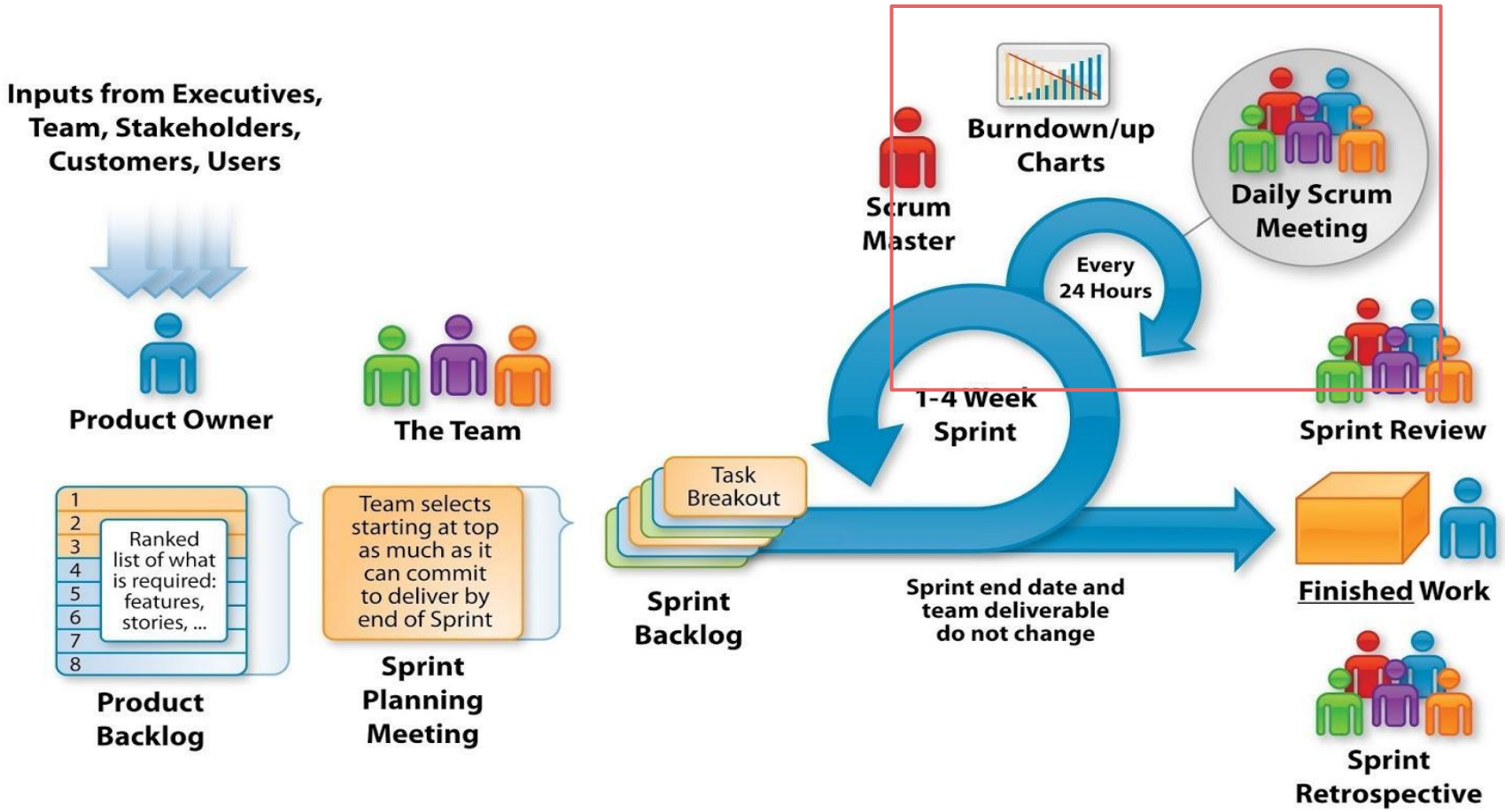
The Agile - Scrum Framework



Artifact - Sprint Board



The Agile - Scrum Framework



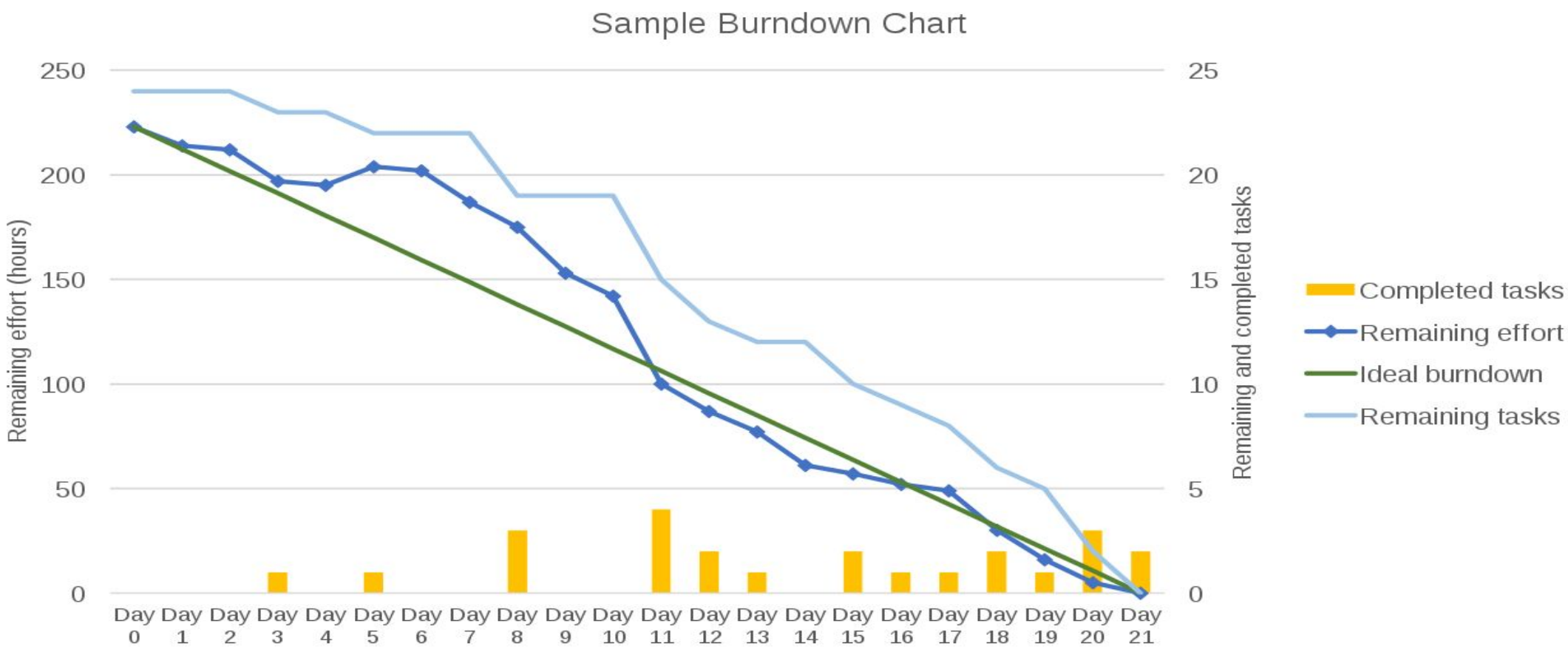
Ceremony - Daily Scrum Meeting

- Parameters
  - Daily, ~15 minutes, Stand-up
  - Anyone late pays a $1 fee
- Not for problem solving
  - Whole world is invited
  - Only team members, Scrum Master, product owner, can talk
  - Helps avoid other unnecessary meetings
- Three questions answered by each team member:
  - What did you do yesterday?
  - What will you do today?
  - What obstacles are in your way?

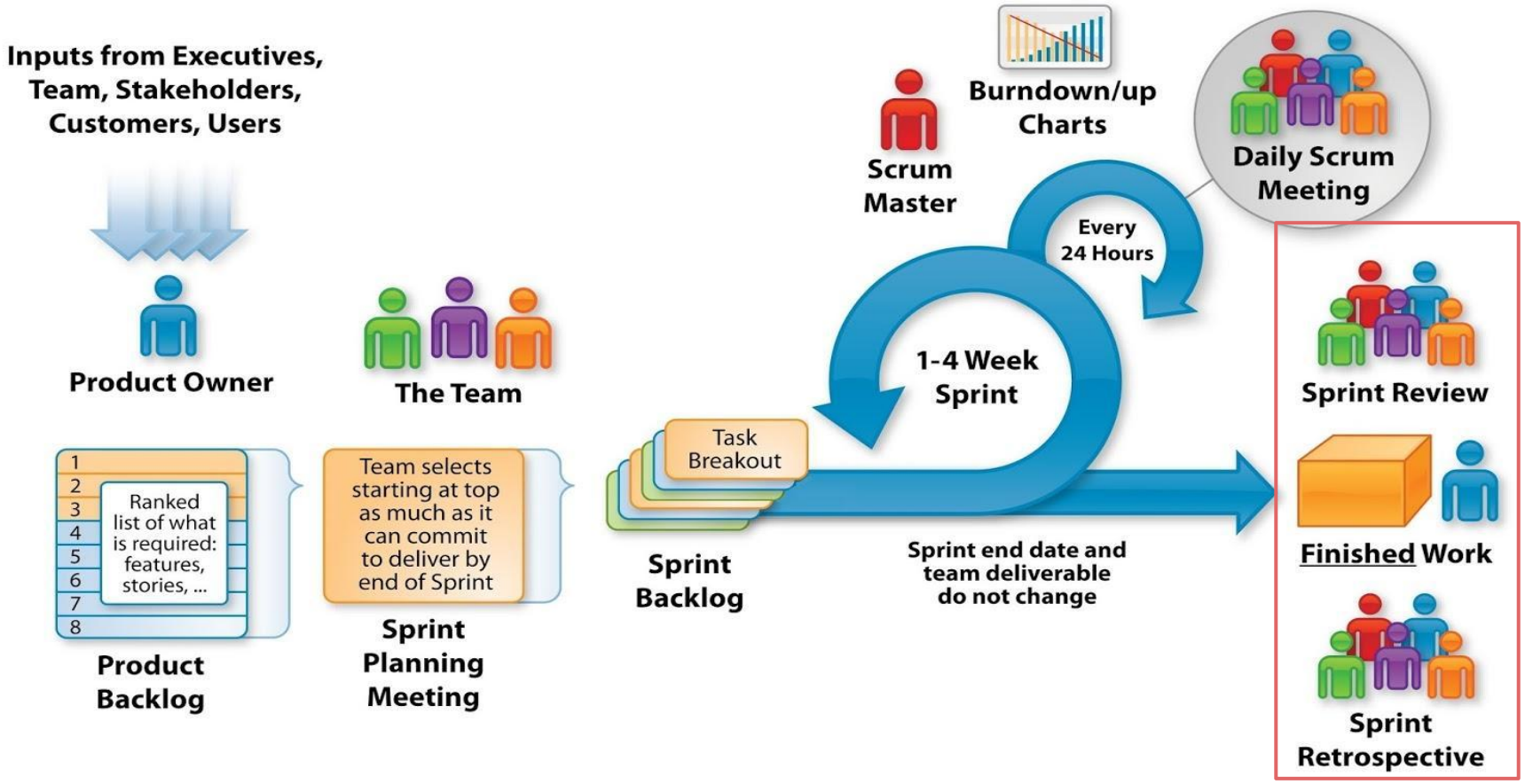
Artifact – Burndown Chart

- A display of what work has been completed and what is left to complete
  - one for each developer or work item
  - updated every day
  - (make best guess about hours/points completed each day)
- Variation: Release burndown chart
  - shows overall progress
  - updated at end of each sprint

Artifact – Burndown Chart



The Agile - Scrum Framework



Ceremony - Sprint Review

- Team presents what it accomplished during the sprint
- Typically takes the form of a demo of new features or underlying architecture
- Informal
  - 2-hour prep time rule
  - No slides
- Whole team participates
- Invite the world

Ceremony - Sprint Retrospective

- Identify the scopes of improvement for better result in next sprints
- What worked well, what went wrong are also discussed

Scrum - When to Use?

- There is a relatively stable, achievable project goal, and the project can be broken into discrete chunks of work.
- The team benefits from regular structured meetings and retrospectives to assess progress and incorporate feedback.
- The organization supports team autonomy and self-organization, and the project can benefit from predictable, iterative progress.

Anthony Sistilli : https://www.youtube.com/@AnthonySistilli

Quiz Announcement

- Date - 10th March
- Topic - SDLC and Agile
- Question Type - Scenario Based Short Questions

Software Engineering Approaches

	Waterfall	V Model	Incremental	Iterative	XP	Scrum	AUP
Small	✅	✅	❌	❌	❌	❌	❌
Medium	❌	✅	✅	✅	❌	✅	✅
Large	❌	❌	✅	✅	✅	✅	✅
Testing	❌	✅	✅	✅	✅	✅	✅
Initial Req. Changing	❌	❌	❌	✅	✅	✅	✅
Process Req. Changing	❌	❌	✅	✅	✅	✅	✅
Early Release	❌	❌	✅	❌	✅	✅	✅
Technologies Changing	❌	❌	❌	✅	❌	✅	✅

Prioritize

Thank you