SOFTWARE PROJECT MANAGEMENT

LECTURE # 4

SOFTWARE DEVELOPMENT FUNDAMENTALS-II

06th October, 2011

Engr. Ali Javed

Contact Information

- Instructor: Engr. Ali Javed

  Lecturer

  Department of Software Engineering

  U.E.T Taxila

  - Email: ali.javed@uettaxila.edu.pk
  - Contact No: +92-51-9047592
  - Office hours:
    - **Monday, 11:00 - 01:00, Office # 7**

Course Information

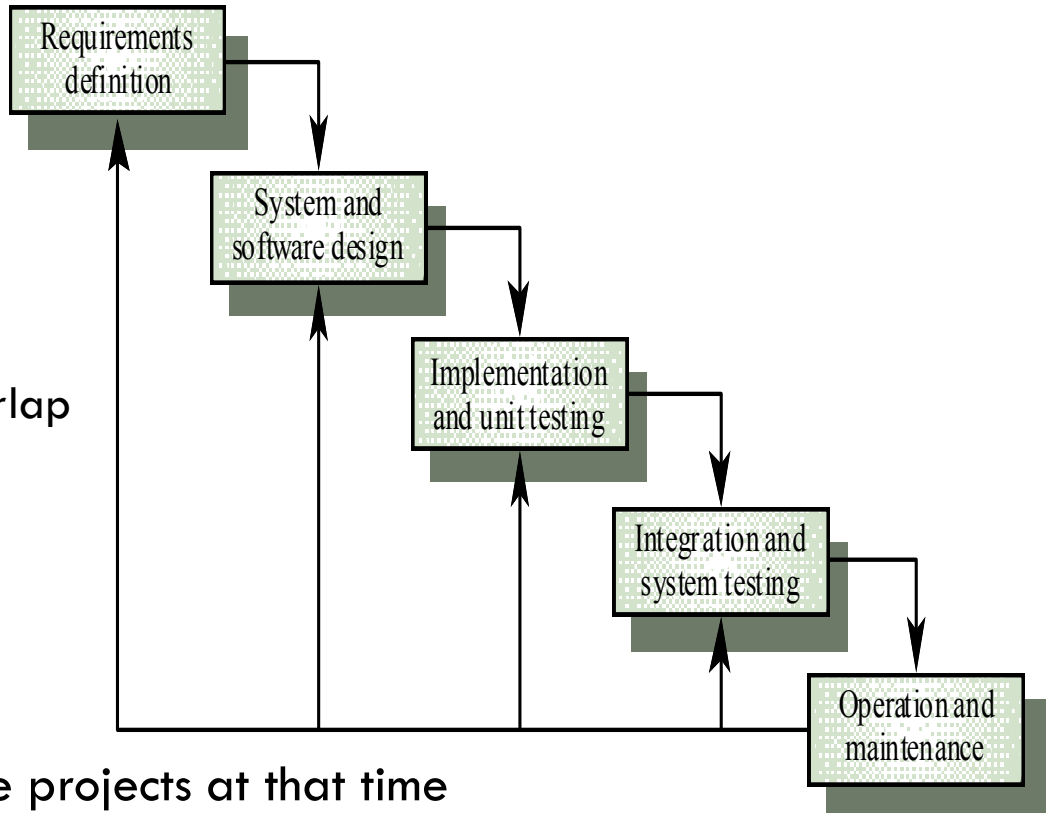
- **Course Name: Software Project Management**
- **Course Code: SE-401**
- **CMS Link:** http://web.uettaxila.edu.pk/CMS/AUT2011/seSPMbs/index.asp

Software Models

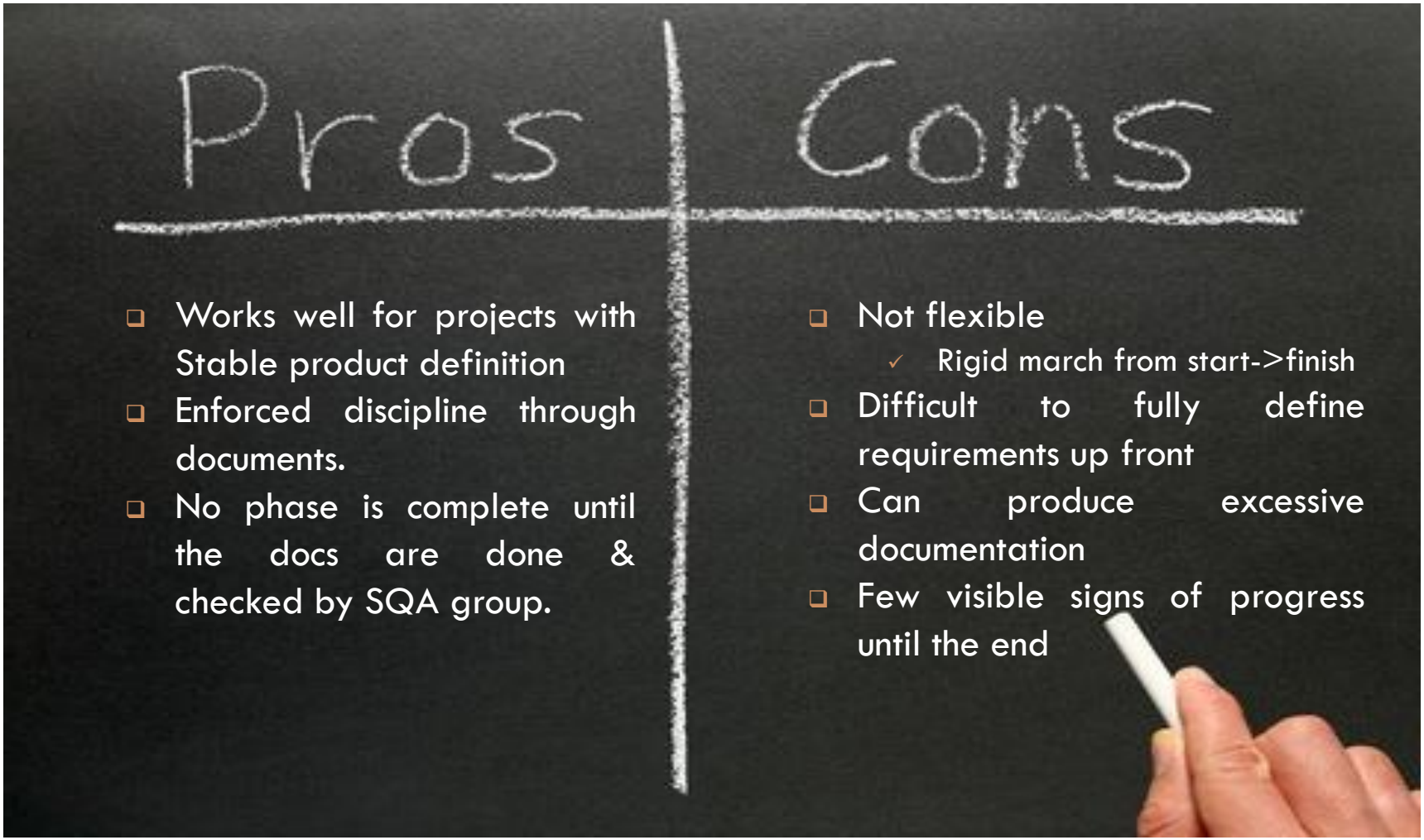
- WaterFall Model
- Evolutionary Development
- Reuse based Development
- Process Iterations
- Agile Development

Pure Waterfall

- Widely used in 80's and 90's
- The "granddaddy" of models
- Linear sequence of phases
  - ✓ "Pure" model: no phases overlap
- Document driven
- All planning done up-front
- Used for medium to large scale projects at that time



Pros and Cons



Evolutionary Development

❑ Exploratory development

- ✓ Objective is to work with customers and to evolve a final system from an initial outline specification.
- ✓ Should start with well-understood requirements.
- ✓ The system evolves by adding new features as they are proposed by customer.

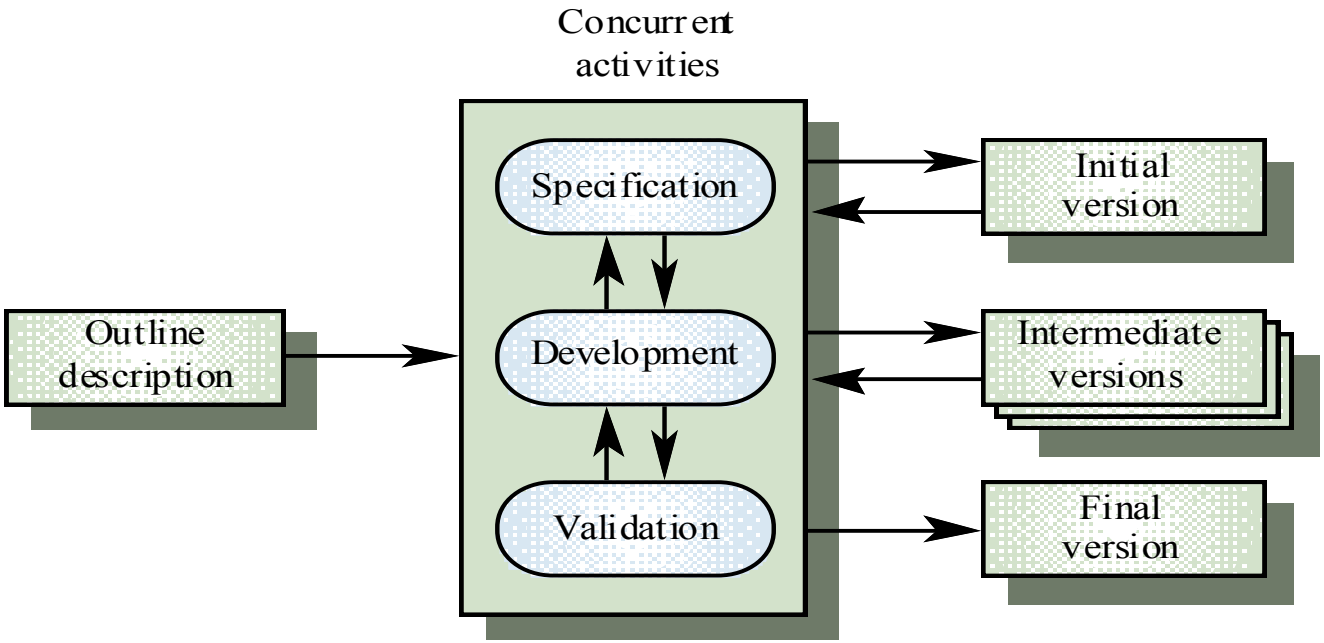
❑ Throw-away prototyping

- ✓ Objective is to understand the system requirements. Should start with poorly understood requirements
  - **Develop "quick and dirty" system quickly;**
  - **Expose to user comment;**
  - **Refine Until adequate system developed.**
- ✓ Particularly suitable where:
  - detailed requirements not possible;
  - powerful development tools (e.g. GUI) available

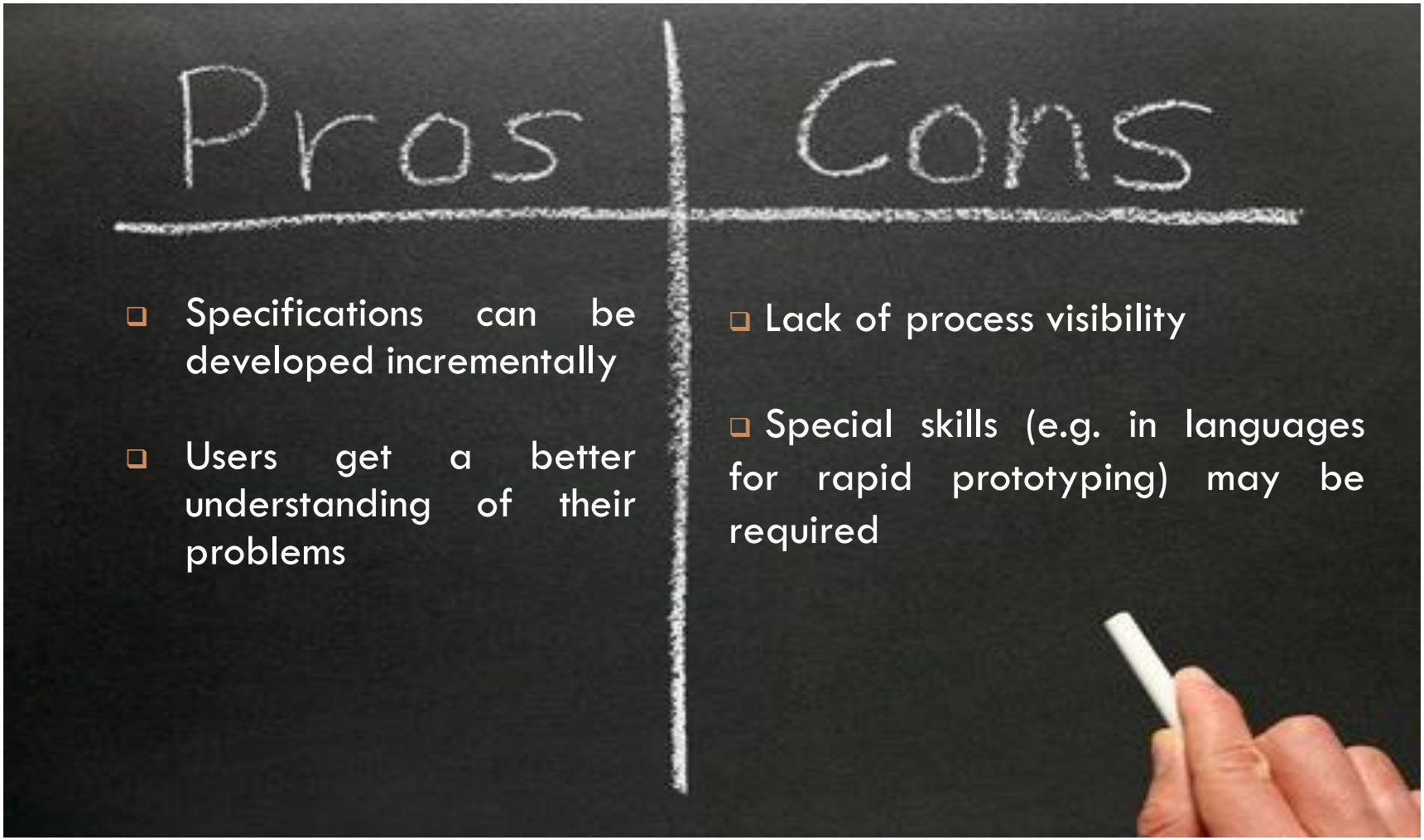
Evolutionary Development

Applicability

- ✓ For small or medium-size systems
- ✓ For parts of large systems (e.g. the user interface)
- ✓ For short-lifetime systems



Evolutionary Development



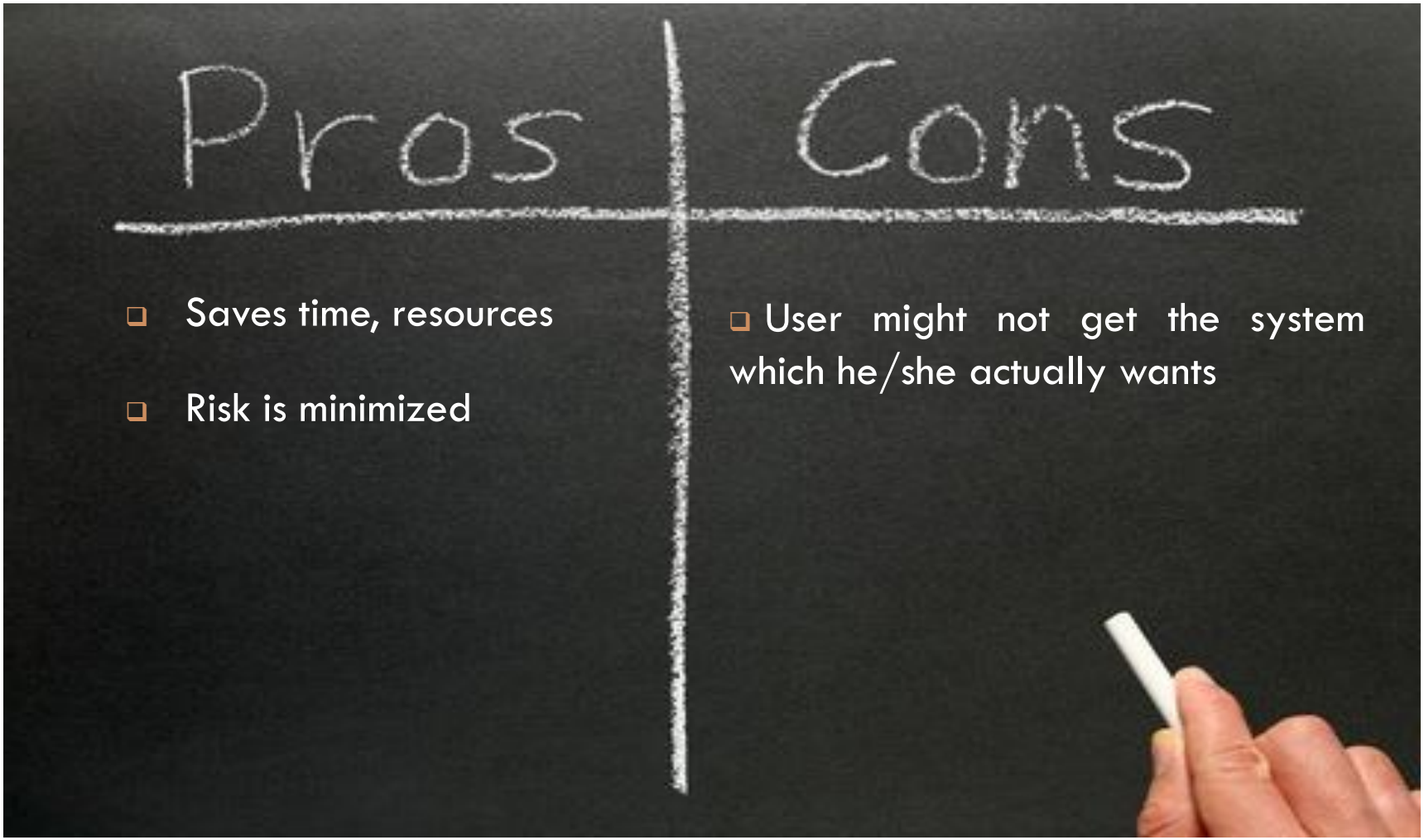
Reuse Based Development

- **Based on systematic reuse** where systems are integrated from existing components or COTS (Commercial-off-the-shelf) systems

- **Process stages**

  - ✓ Component analysis
  - ✓ Requirements modification
  - ✓ System design with reuse
  - ✓ Development and integration

Reuse Based Development

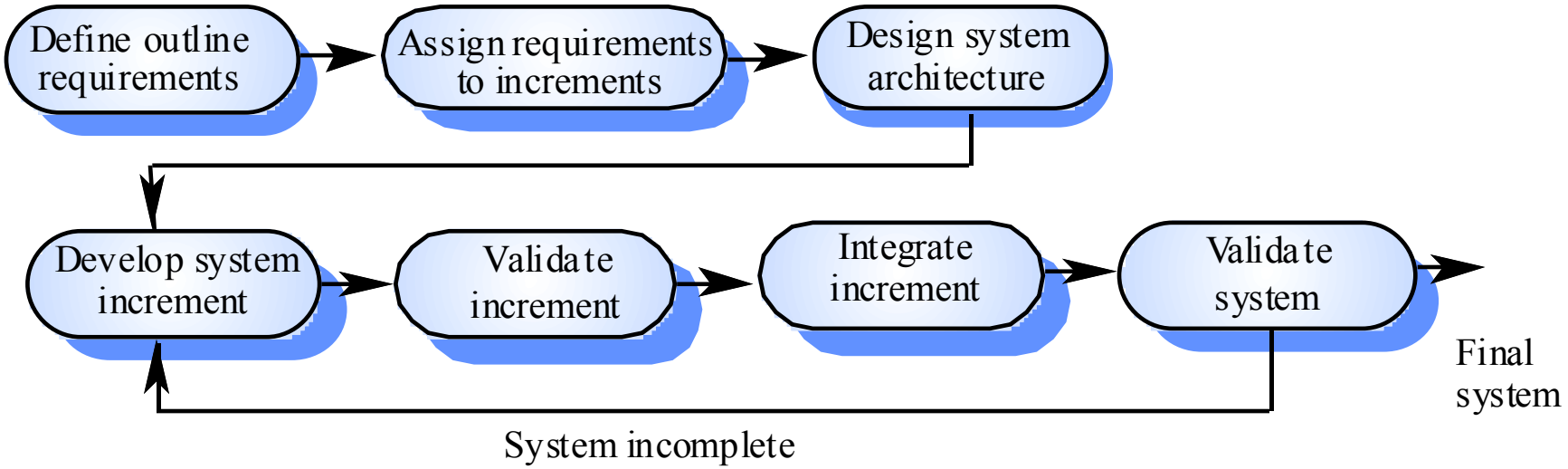


Process Iteration

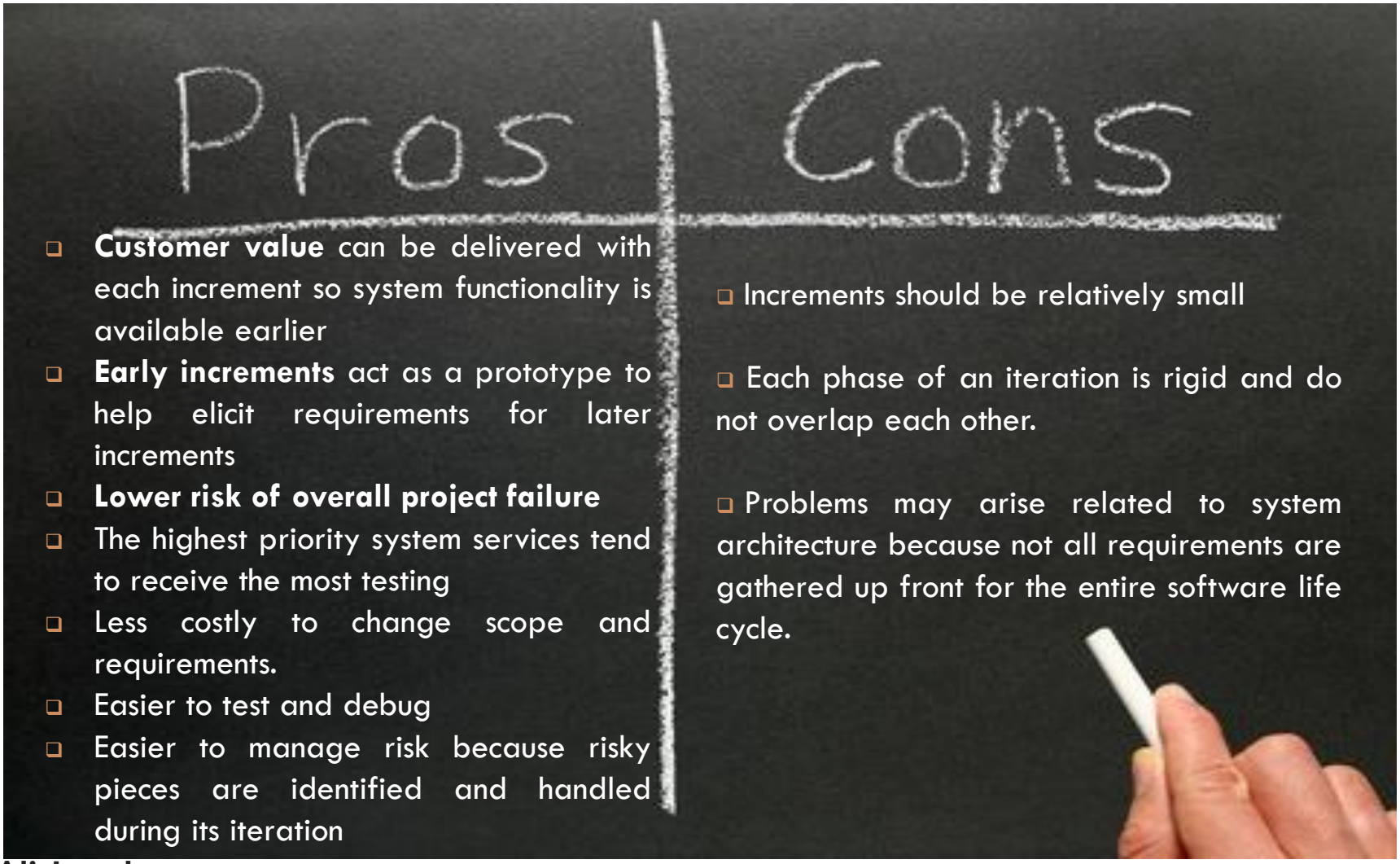
- **Modern development processes take iteration as fundamental,** and try to provide ways of managing, rather than ignoring, the risk
- **System requirements ALWAYS evolve in the course of a project** so process iteration where earlier stages are reworked is always part of the process for large systems
- **Iteration** can be applied to any of the generic process models
- **Two (related) approaches**
  - ✓ **Incremental development**
  - ✓ **Spiral development**

Incremental Development

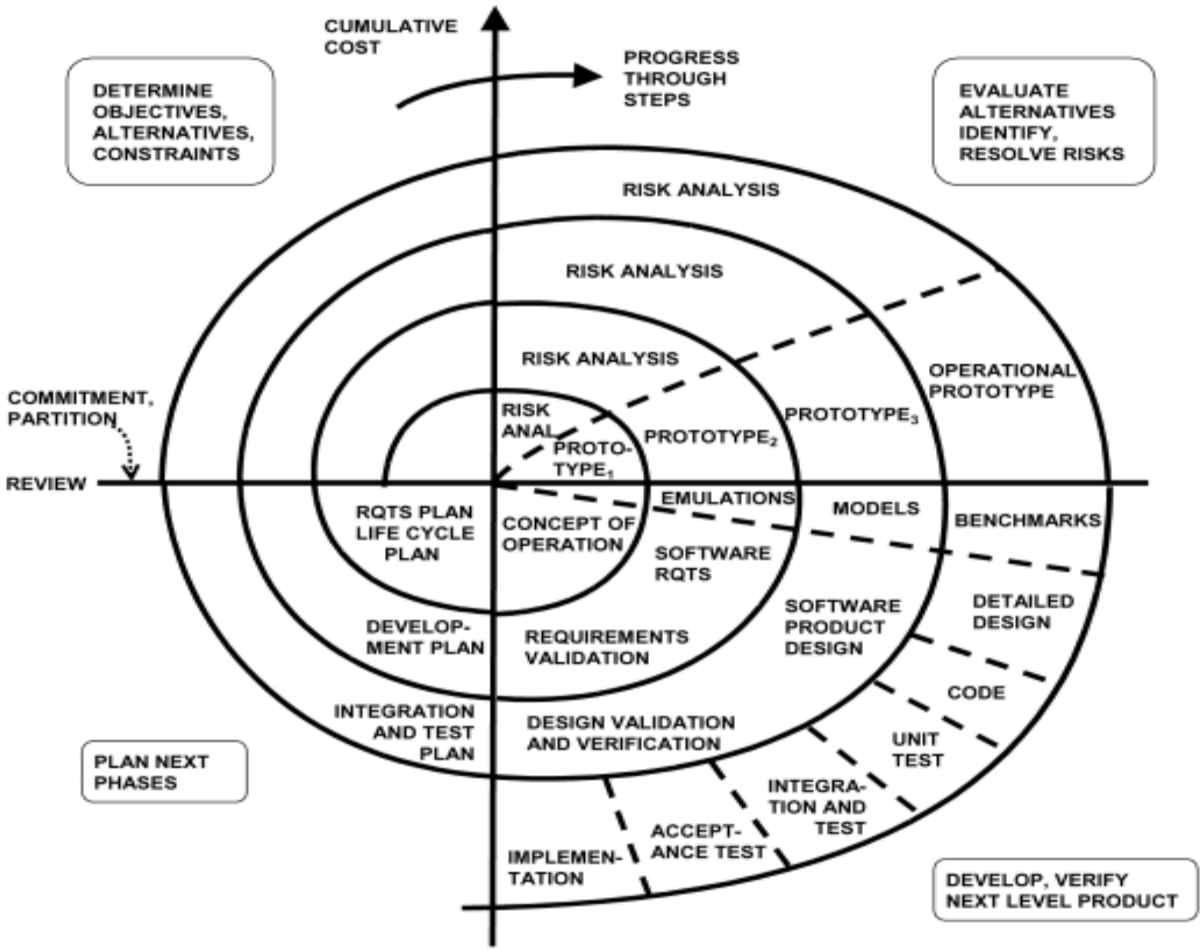
- Rather than deliver the system as a single delivery, **the development and delivery is broken down into increments** with each increment delivering part of the required functionality
- **User requirements are prioritised** and the highest priority requirements are included in early increments
- **Once the development of an increment is started, the requirements are frozen** though requirements for later increments can continue to evolve



Incremental Development [3]

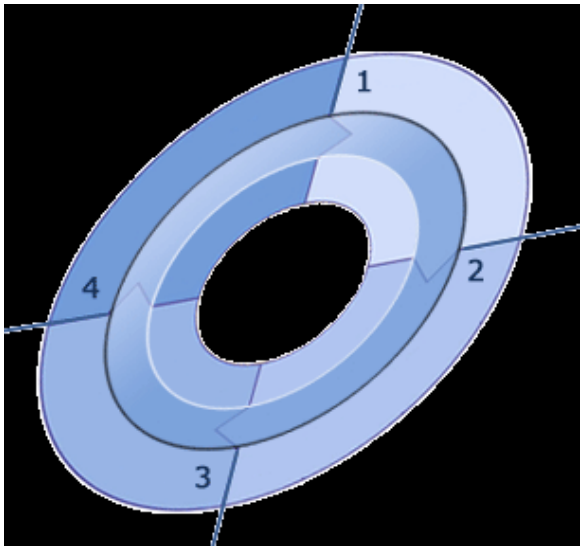


Spiral Model



Spiral Model

- **Process is represented as a spiral** rather than as a sequence of activities with backtracking
- **Each loop in the spiral represents a phase** in the process.
- **No fixed phases such as specification or design** - loops in the spiral are chosen depending on what is required
- **Risks are explicitly assessed and resolved** throughout the process



Spiral Model

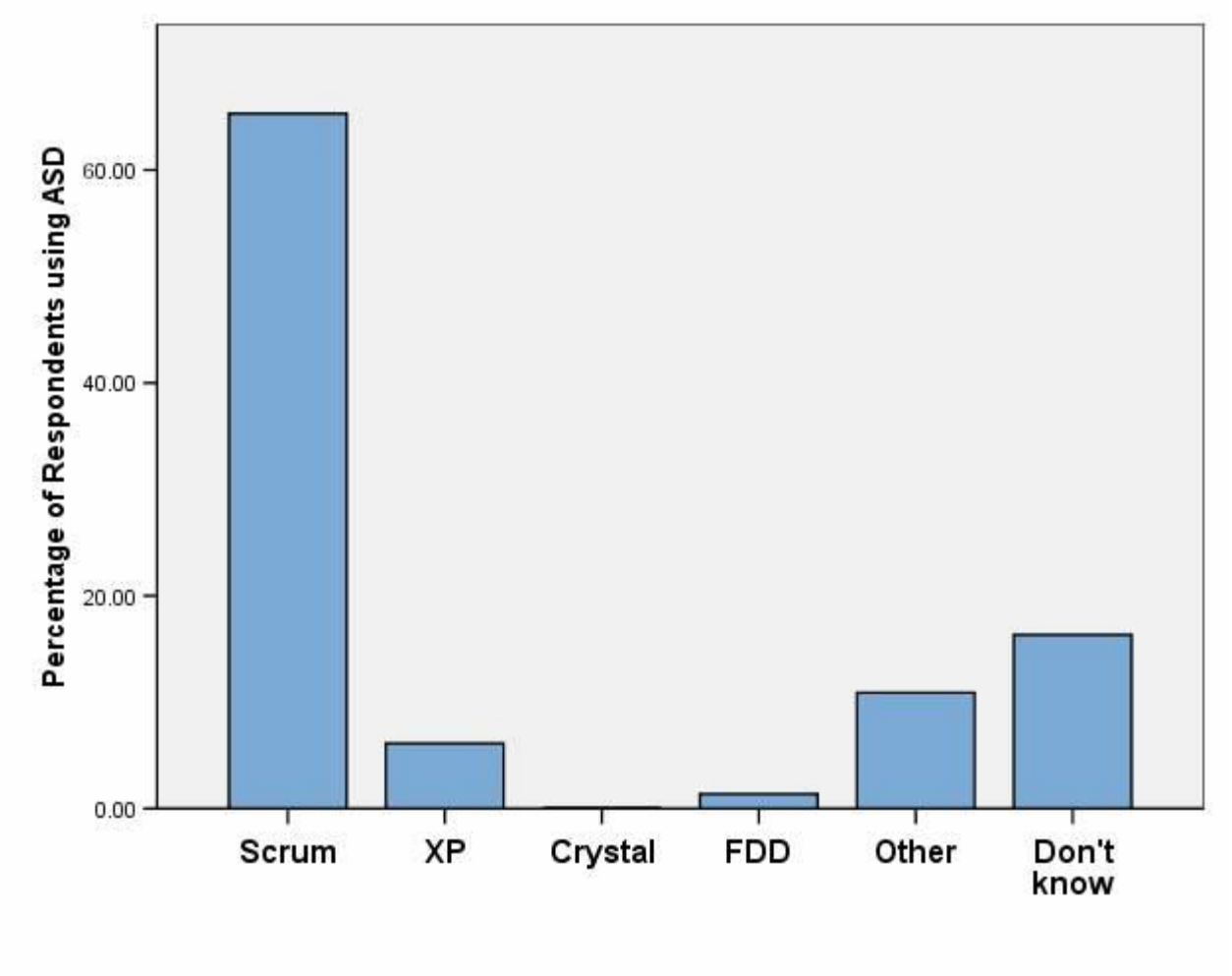
- **Objective setting**
  - ✓ Specific objectives for the phase are identified
- **Risk assessment and reduction**
  - ✓ Risks are assessed and activities put in place to reduce the key risks
- **Development and validation**
  - ✓ A development model for the system is chosen which can be any of the generic models
- **Planning**
  - ✓ The project is reviewed and the next phase of the spiral is planned

What is Agile?

- **Agile software development** is a group of software development methodologies based on iterative and incremental development, where requirements and solutions evolve through collaboration between self-organizing, cross-functional teams. It promotes adaptive planning, evolutionary development and delivery; time boxed iterative approach and encourages rapid and flexible response to change.

- **Common Practices**
  - ✓ Regular Deployment of Working Software
  - ✓ Pair Programming
  - ✓ Refactoring [14]
  - ✓ Continuous Integration
  - ✓ Test Driven Development (TDD)
  - ✓ Shared Code Ownership
  - ✓ Active Stakeholder Participation

Agile Methodology



Some Agile Methodologies

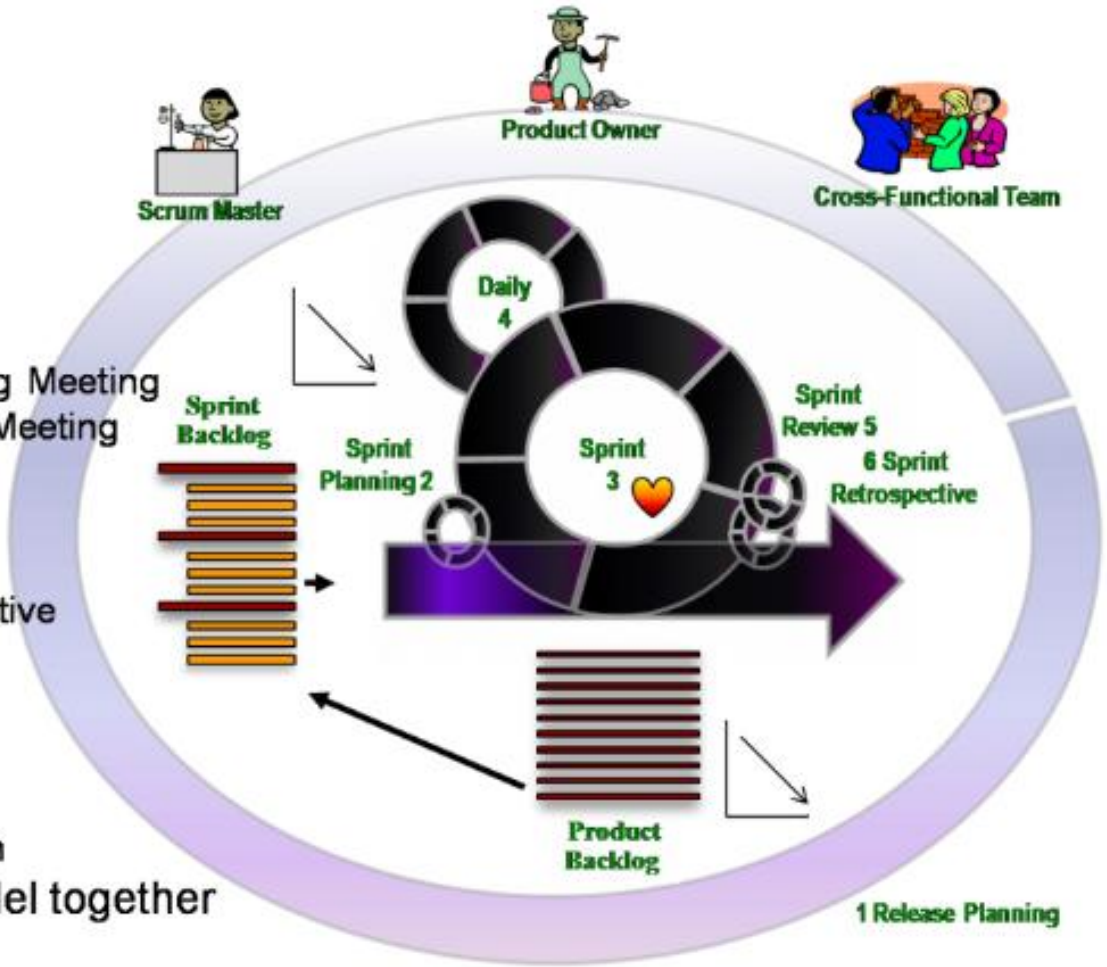
- Scrum
- Extreme Programming (XP)
- Feature-Driven Development (FDD)
- Adaptive Software Process
- Crystal Light Methodologies
- Dynamic Systems Development Method (DSDM)
- Lean Development

SCRUM [10,11]

- **Scrum** is an iterative, incremental framework for project management often seen in agile software development
- It defines a set of activities that can help your team deliver more value to your customers faster.
- These activities provide your customers with the opportunity to review, guide and influence your team's work as it progresses.
- This approach does not attempt to define everything at the start of a project. Instead, your team works in short iterations (also called sprints) and refines the plan as the team makes progress.

A Model of SCRUM

- Roles
  - ScrumMaster
  - Team
  - Product Owner
- Times Boxes
  1. Release Planning Meeting
  2. Sprint Planning Meeting
  3. Sprint
  4. Daily Scrum
  5. Sprint Review
  6. Sprint Retrospective
- Artifacts
  - Product Backlog
  - Sprint Backlog
  - Sprint Burndown
  - Release Burndown
- Rules bind the model together



Scrum Roles

- Scrum Master
- Team
- Product Owner

The SCRUM Master [3,5]

- In the Scrum process, Scrum Master has a role of coach , fixer and gatekeeper
- The job of the scrum master is to make sure that the project is progressing smoothly
- He sets the meetings, monitor the work and facilitate release planning
- Two important task of scrum master are:
  - ✓ Protecting the team from outside disturbance
  - ✓ Clears the ways for the team by helping them to solve their problems

The SCRUM Team [4]

- In Scrum, an ideal team would include seven members, plus or minus two. Usually, teams are comprised of cross-functional members, including software engineers, architects, programmers, analysts, QA experts, testers, UI designers, etc. It is recommended all team members be located in the same room, called the team room.
- the team has the autonomy to determine how and when to complete its work. As long as the team finishes its work by the deadline and under budget, it is entirely up to the team to determine how that happens.

The Product Owner [6]

- In Scrum, the Product Owner is the one person responsible for a project's success.
- The Product Owner outlines work in the Product backlog
- Product Owner makes sure that right features to be included in the product backlog
- Of course, he or she must also consider the stakeholders (to make sure their interests are included in the release) and the team (to make sure the release is developed by the deadline and within budget).

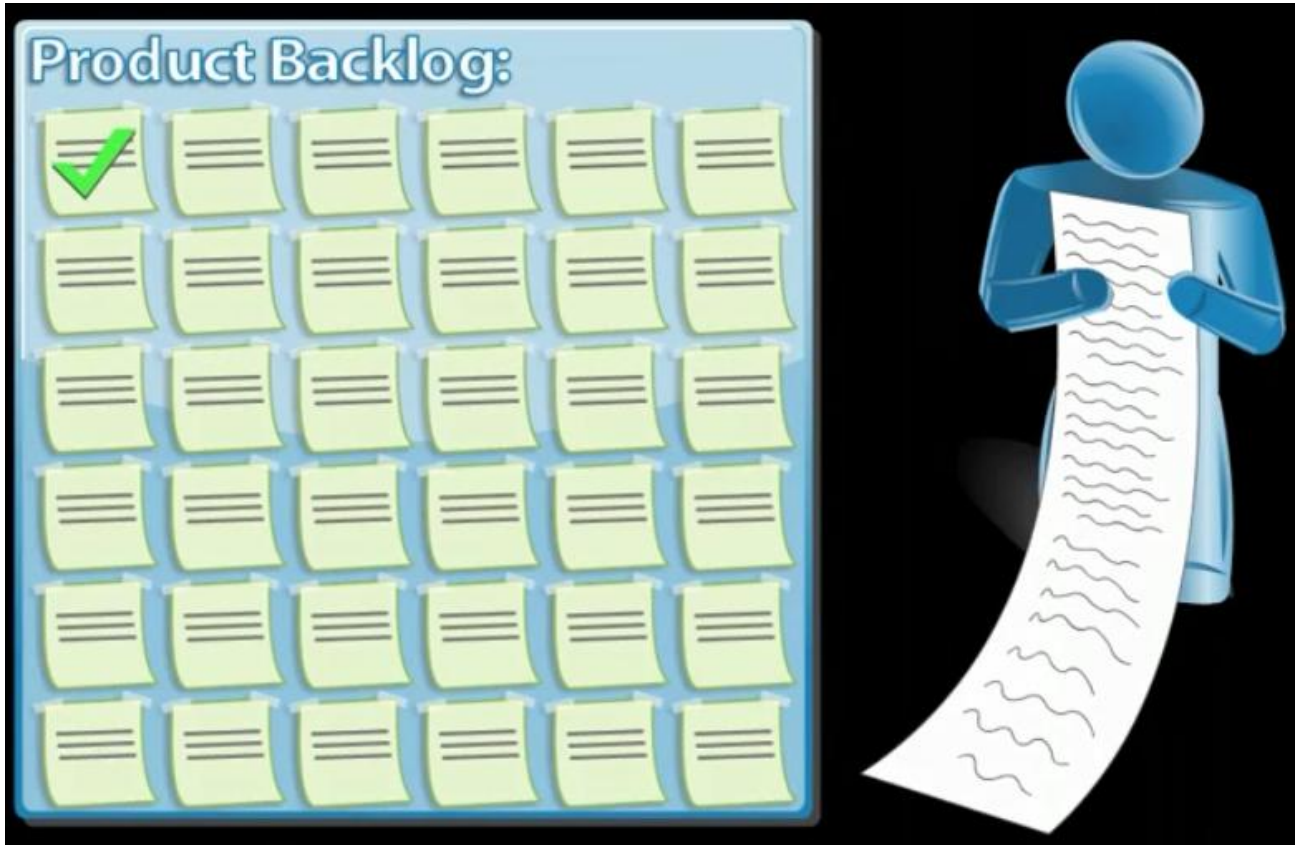
The Product Owner [6]

Scrum Artifacts

- Product Backlog
- Sprint Backlog
- Sprint
- Burn down Chart

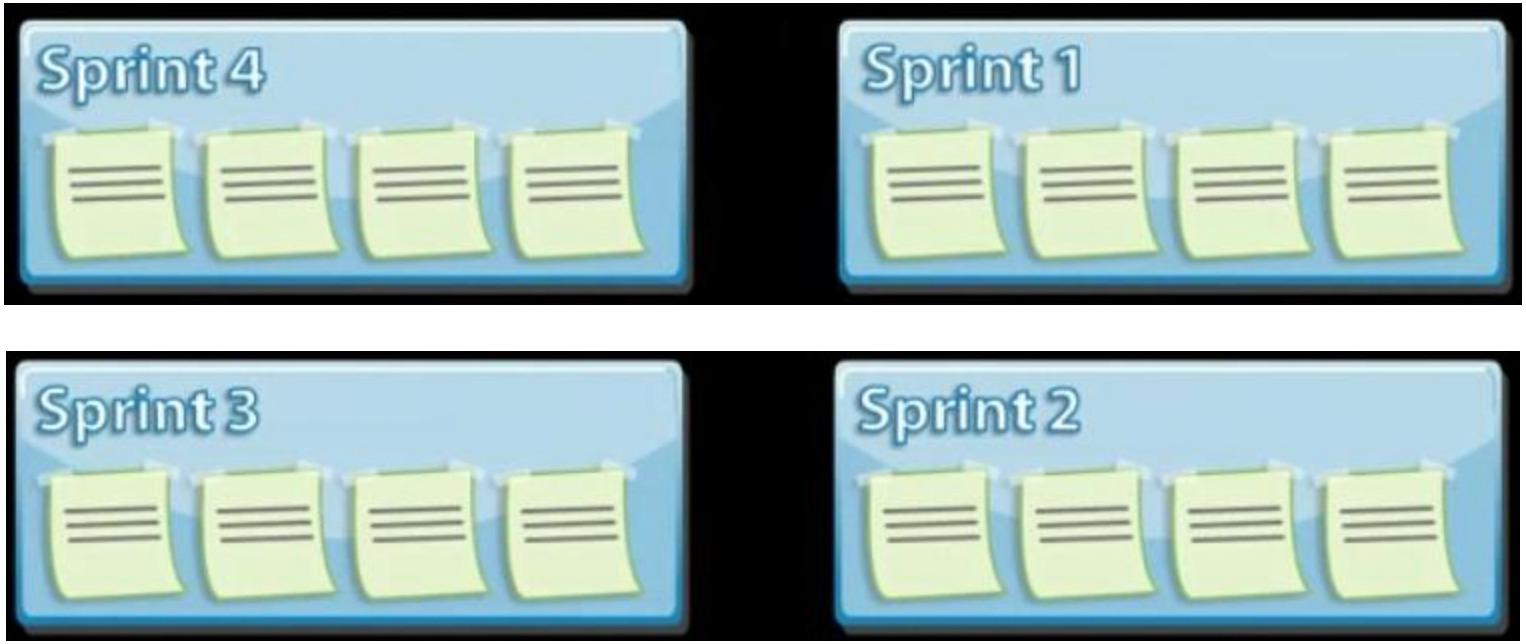
Product Backlog

- Contains all the currently known requirements for a product
- Is managed by the Product Owner and can change as needed



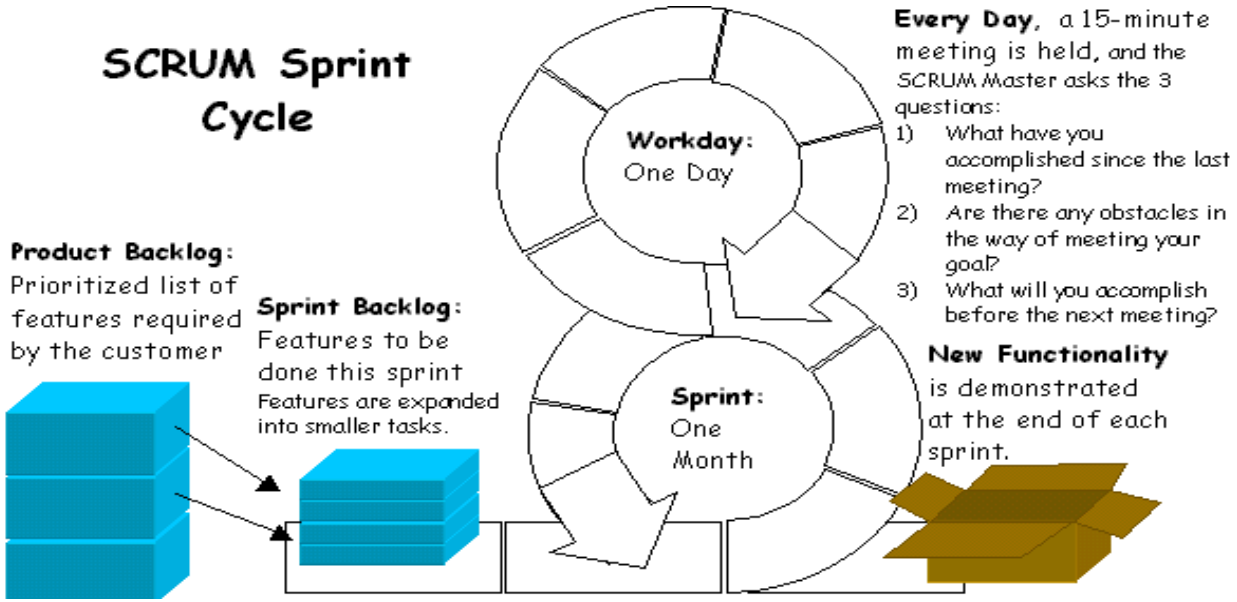
Sprint Backlog

- Contains the set of prioritized Product Backlog items that are currently being worked on



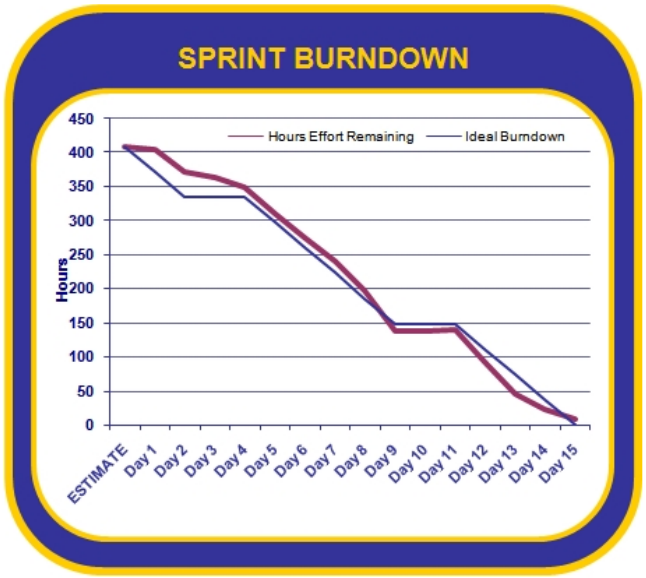
Sprint [7]

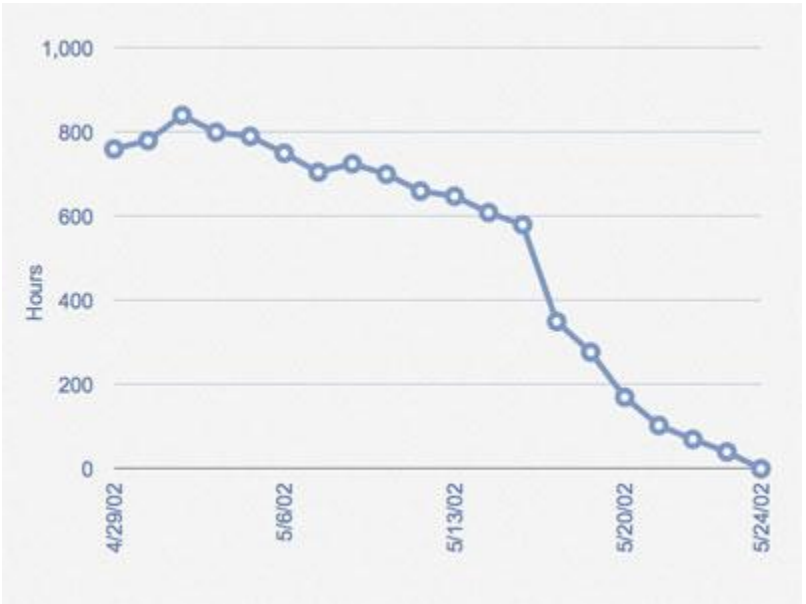
- The product is developed in a series of 1-to-4-week iterations, or sprints. Before a sprint is begun, a Sprint Planning Meeting is held to determine what features are to be implemented in that sprint. The sprint has 4 major steps:
  - ✓ **Develop** the product further - implement, test, and document.
  - ✓ **Wrap** up the work - get it ready to be evaluated and integrated.
  - ✓ **Review** the work done in this sprint.
  - ✓ **Adjust** for any changes in requirements or plans.
- Results in an incremental delivery of usable product



Sprint Burn down Chart [9]

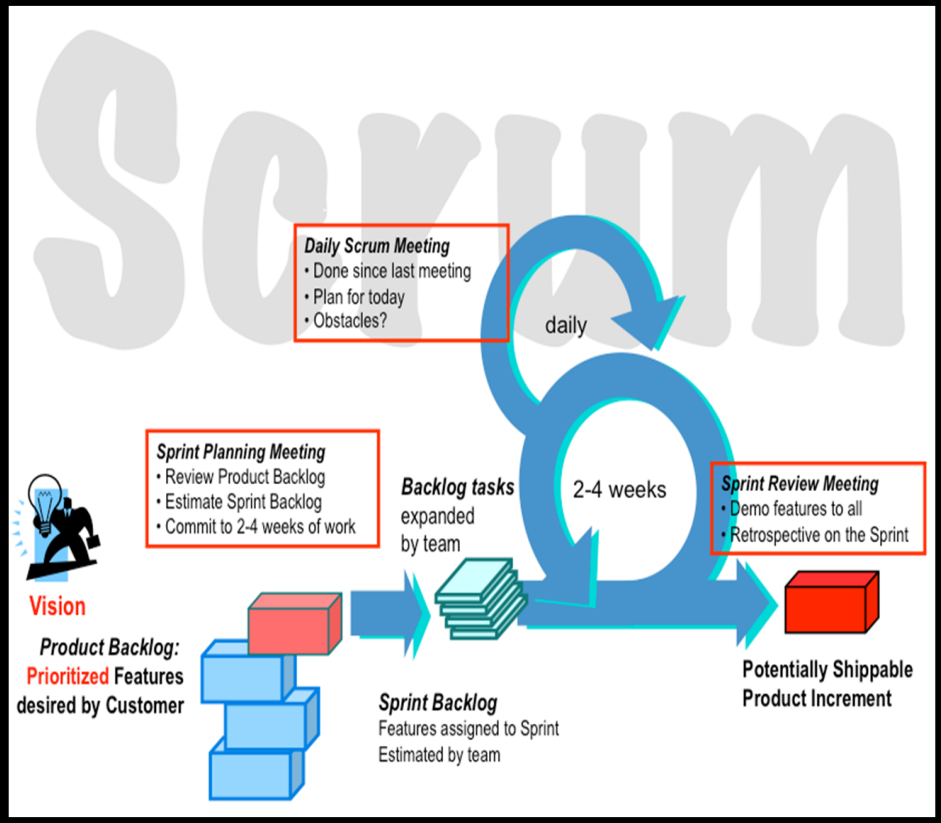
- The estimated work remaining in the sprint is calculated daily and graphed, resulting in a Sprint Burn down Chart
- The vertical axis displays the hours of effort remaining for the Sprint.
- The horizontal axis displays the days of the Sprint.
- The burn down is supposed to be shown by the line of descent from the start of the Sprint with the starting hours, down to the end of the Sprint with no hours remaining.



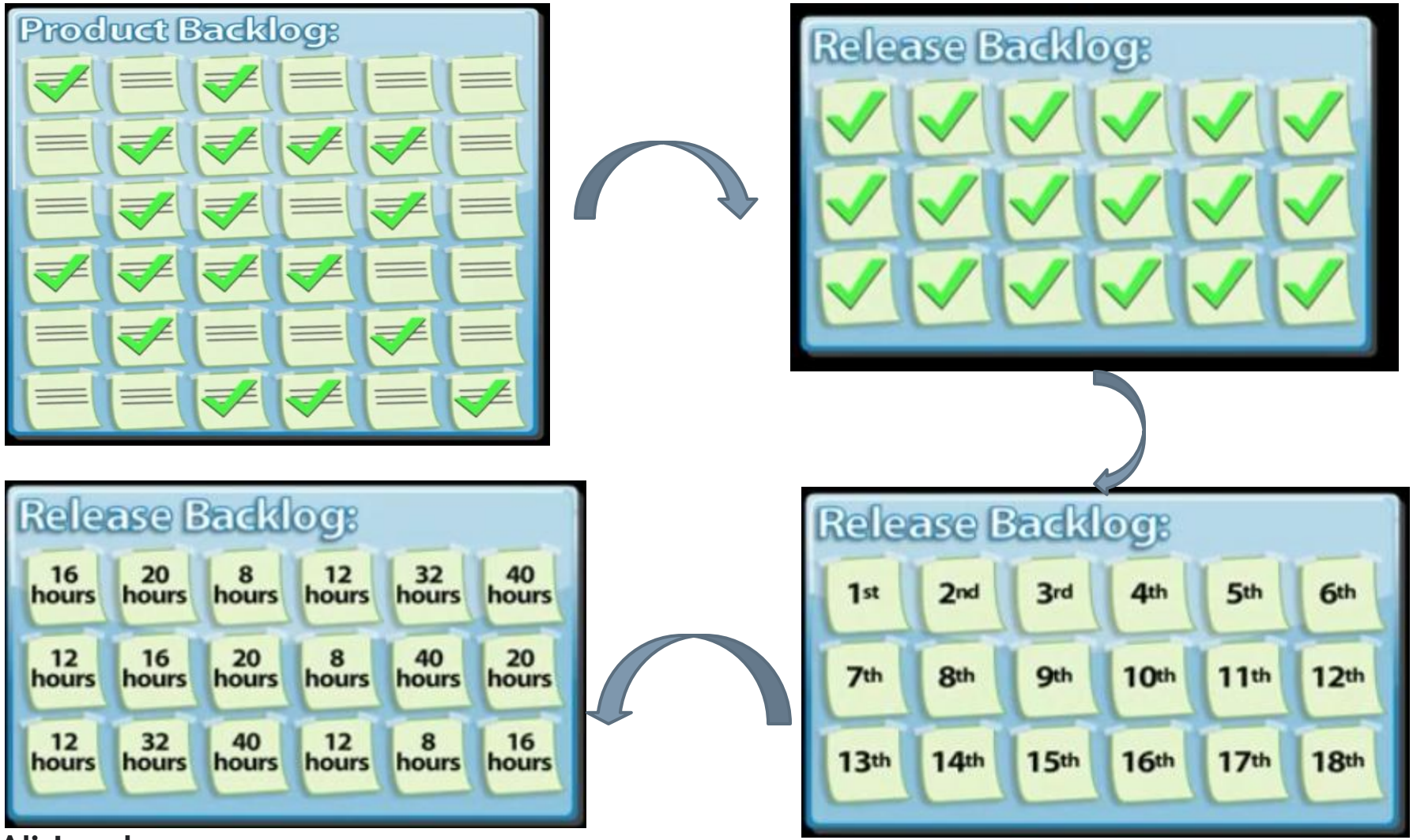


Scrum Meetings

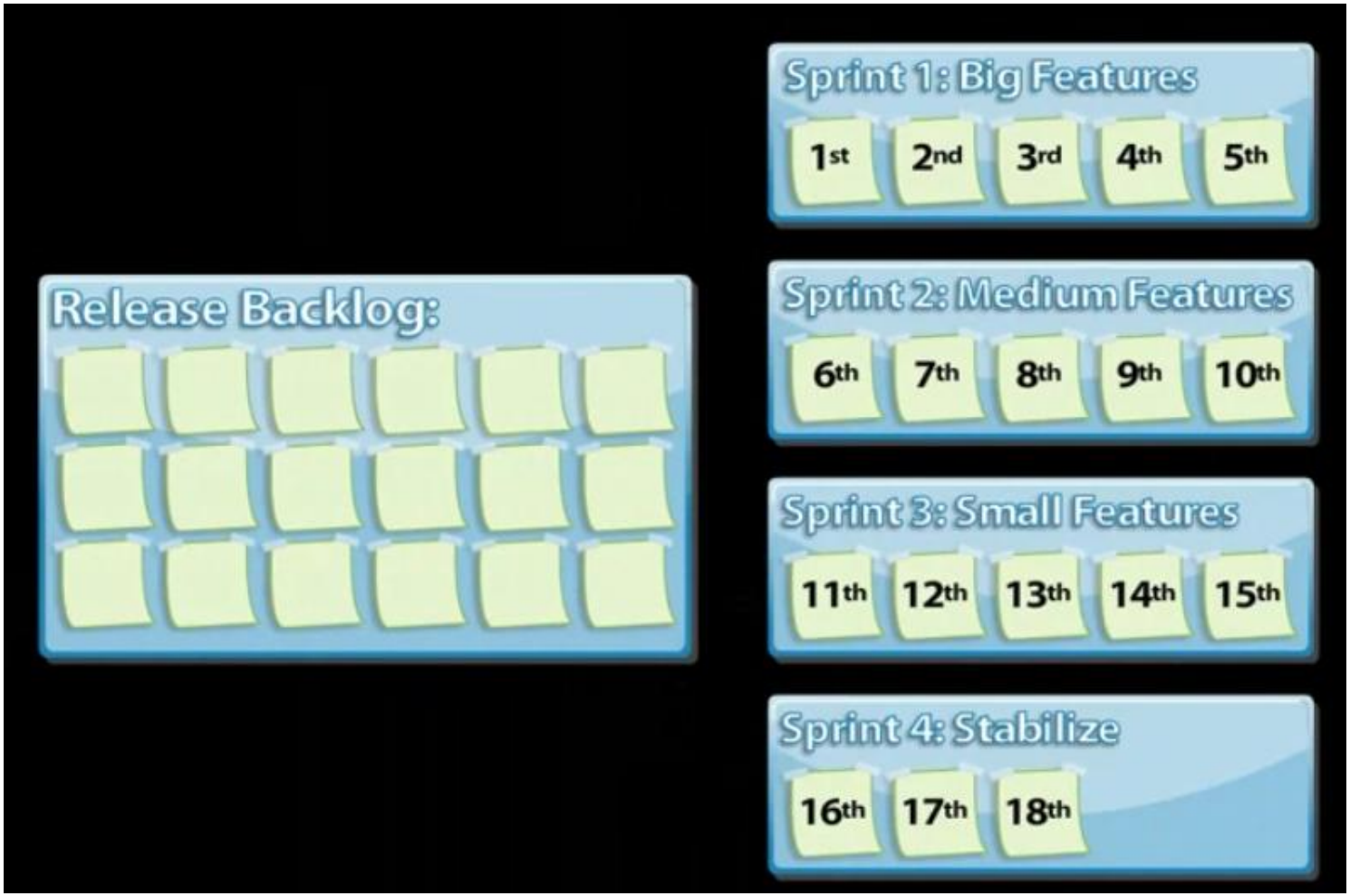
- Release Planning Meeting
- Sprint Planning Meeting
- Sprint Review Meeting
- Sprint Retrospective Meeting
- Daily Scrum Meeting



Release Planning



Release Planning



Sprint Planning Meeting

- A meeting at the beginning of a sprint where the sprint is planned.
- Items from the Product Backlog are selected to be completed in the sprint, based on the priorities set by the Product Owner. Eight hour time limit
  - ✓ (1st four hours) Product Owner + Team: dialog for prioritizing the Product Backlog
  - ✓ (2nd four hours) Team only: hashing out a plan for the Sprint, resulting in the Sprint Backlog

Sprint Review Meeting [7]

- Review the work that was completed and not completed
- Present the completed work to the stakeholders (a.k.a. “the demo”)
- Incomplete work cannot be demonstrated
- Four hour time limit

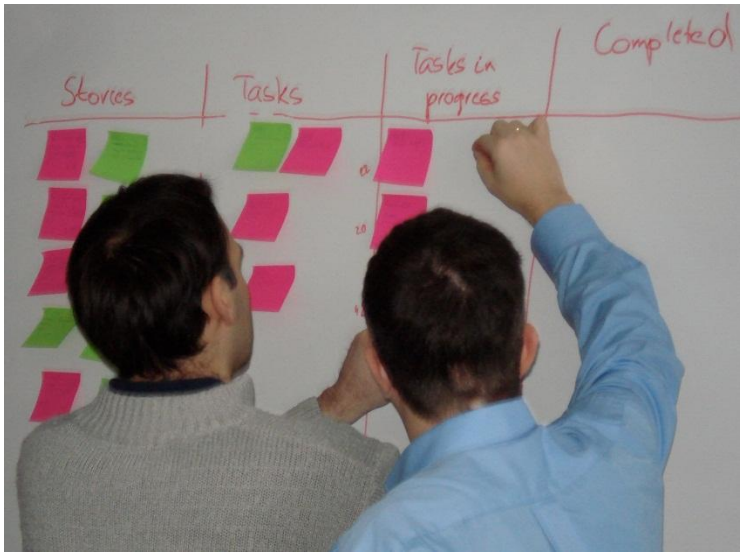
Sprint Retrospective Meeting [8]

- The sprint retrospective meeting is time boxed to 3 hours.
- It is attended only by the team, the scrum master and the product owner. The product owner is optional.
- **Make continuous process improvements**
- Start the meeting by having all team members answer two questions;
  - ✓ What went well during the sprint?
  - ✓ What could be improved in the next sprint?

Daily SCRUM Meeting

- Brief 'Stand-up' meeting each morning with SCRUM Team only
- Duration is 15 min
- Three questions are asked
  - ✓ What value did you add yesterday?
  - ✓ What value will you add today?
  - ✓ What will stop you making progress?

Team members in the scrum meeting



References

1. Software Engineering by Roger Pressman
2. Software Engineering by Ian Sommerville
3. http://3back.com/scrum/certified-scrummaster-training/
4. http://scrummethodology.com/the-scrum-team-role/
5. YouTube - Scrum Master in Under 10 Minutes (HD) by @hamids.wmv
6. http://scrummethodology.com/scrum-product-owner/
7. http://www.codeproject.com/KB/architecture/scrum.aspx
8. http://abrachan.net/scrum/scrum/sprint-retrospective-meeting/
9. http://epf.eclipse.org/wikis/scrum/Scrum/workproducts/sprint_burndown_chart_F647C347.html
10. http://en.wikipedia.org/wiki/Scrum_%28development%29
11. http://msdn.microsoft.com/en-us/library/dd997796.aspx
12. http://en.wikipedia.org/wiki/Test_harness
13. http://en.wikipedia.org/wiki/Timeboxing
14. http://en.wikipedia.org/wiki/Code_refactoring
15. http://en.wikipedia.org/wiki/Adaptive_Software_Development
16. http://productdevelop.blogspot.com/2011/07/what-is-adaptive-software-development.html

References

17. http://en.wikipedia.org/wiki/Feature-driven_development
18. http://en.wikipedia.org/wiki/Lean_software_development
19. http://www.lean.org/whatslean/
20. http://cseweb.ucsd.edu/users/wgg/CSE131B/Design/node6.html
21. http://en.wikipedia.org/wiki/Extreme_Programming
22. http://www.learnmanagement2.com/flat%20structure.htm
23. http://books.google.com.pk/books?id=8o1eom6ifIMC&pg=PA17&lpg=PA17&dq=difference+between+perceived+integrity+and+conceptual+integrity&source=bl&ots=m1VuEK93KM&sig=5TcbxyjCdO57OJ7VIvC4TLk3ELI&hl=en&ei=VXmWTruiLITKhAeng62EBA&sa=X&oi=book_result&ct=result&resnum=3&ved=0CC0Q6AEwAg#v=onepage&q=difference%20between%20perceived%20integrity%20and%20conceptual%20integrity&f=false

For any query Feel Free to ask