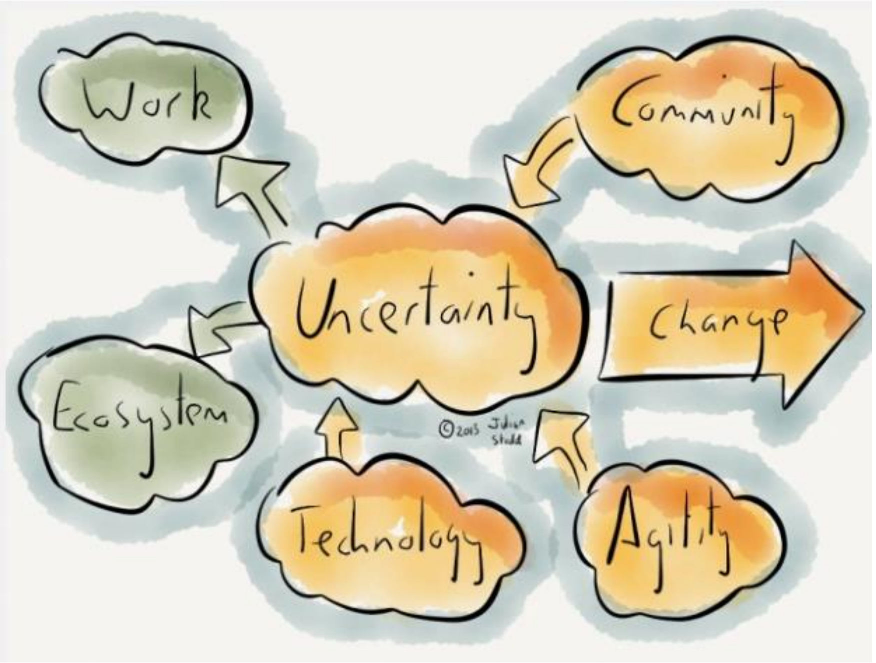
Agile Software Project Management

Definable Work to High Uncertainty Work

Project work ranges from Definable Work to High Uncertainty Work.

Definable Work: Definable work projects are characterized by clear procedures that have proved successful on similar project in the past. Few example of Definable work are Car Production, Microwave Oven or Home after the design is complete. In definable work uncertainty is very less. More definable work is automated.

Uncertainty Work: This include work with high rates of change, complexity and risk. High Uncertainty work include work of System engineer, doctors, teachers and lawyers.



[Uncertainty is the only constant]

These characteristics can present problems for traditional predictive approaches that aim to determine the bulk of the requirements upfront and control changes through a change request process.

From Traditional to Agile

Traditionally waterfall development activities proceed sequentially, from conceptualization of the problem to delivery of the final product. Its also called **linear-sequential life cycle model** Each activity of the process takes as input the outputs produced by the previous activity

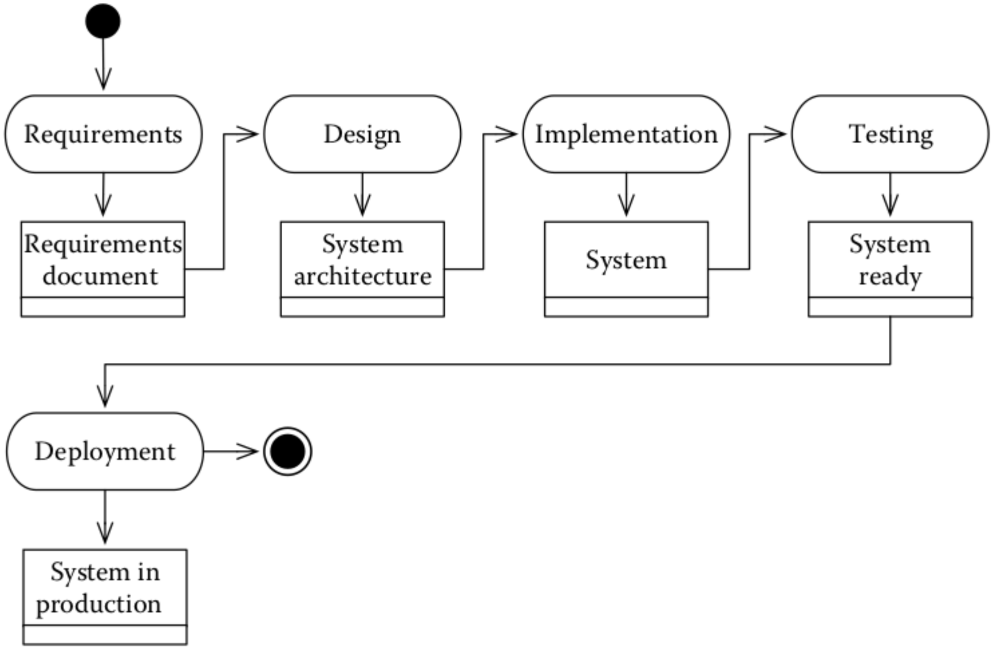


Figure 7.1 The waterfall.

ISSUES

1 High amounts of risk and uncertainty.

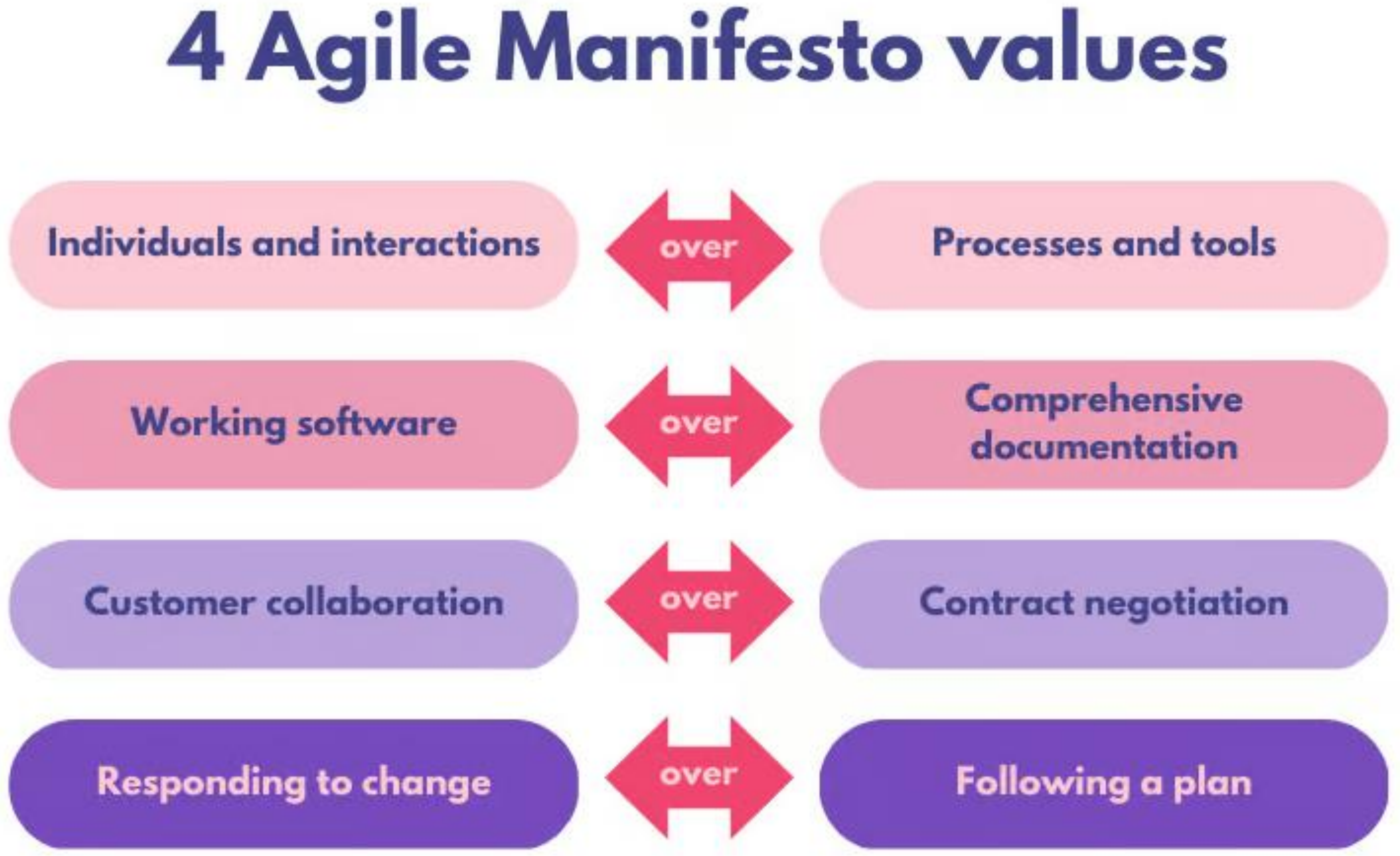
2 Not a good model for complex and object-oriented projects.

3 Poor model for long and ongoing projects.

4 Not suitable for the projects where requirements are at a moderate to high risk of changing. So, risk and uncertainty is high with this process model.

5 It is difficult to measure progress within stages.

The Agile Manifesto and Mindset



4 Agile Manifesto values

Individuals and interactions over Processes and tools

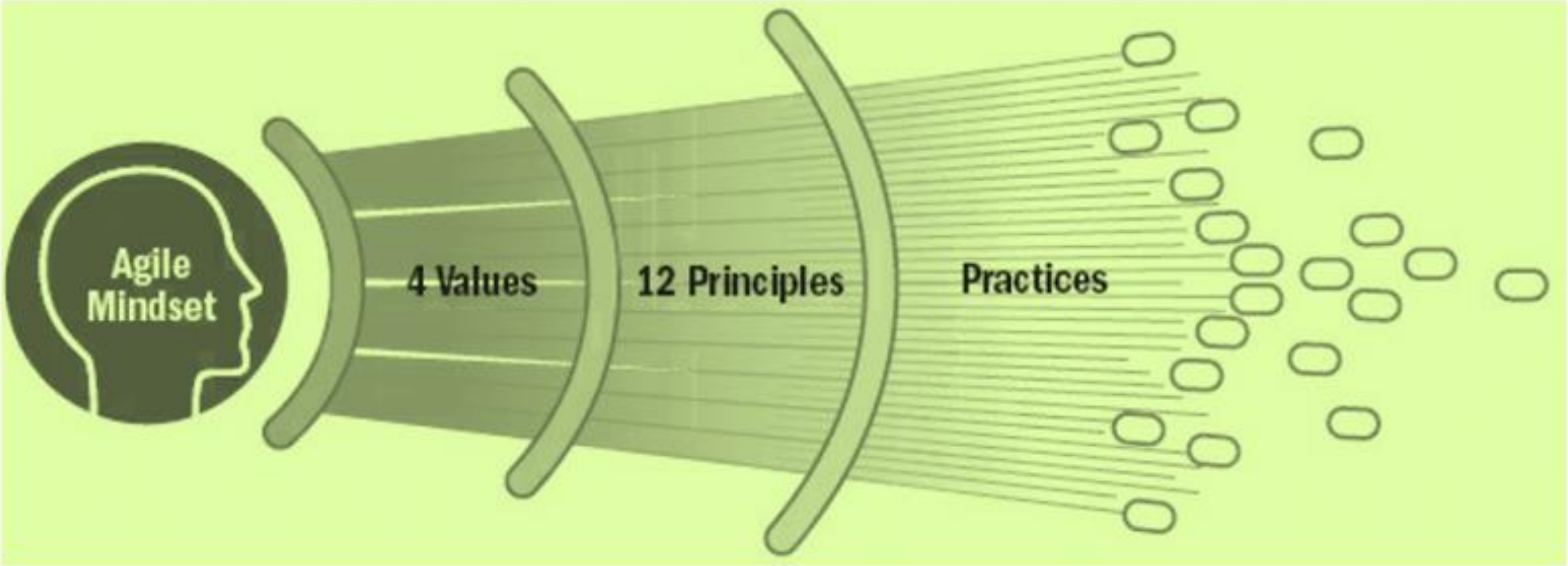
Working software over Comprehensive documentation

Customer collaboration over Contract negotiation

Responding to change over Following a plan

Agile is a mindset defined by **values**, guided by **principles**, and manifested through many different **practices**

Agile practitioners select practices based on their needs

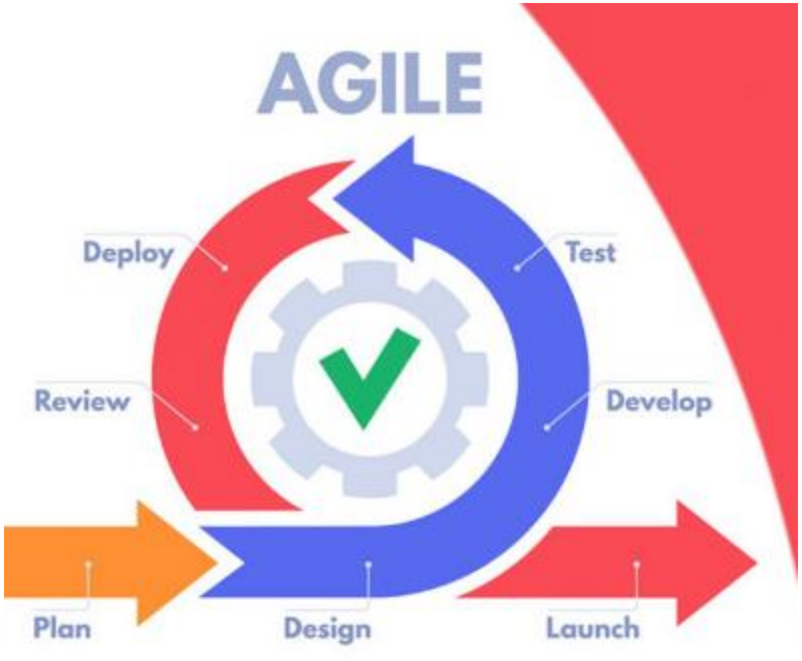


Agile?

- Agile is used to denote the ability of ***Agile*** Methods to respond to changing requirement in a ***controlled*** but ***flexible*** manner
  - Agile methodologies can equip experienced Project Managers with new tools to manage projects that are set in environments of *constant change*.

Agile

Agile project management is an **iterative approach** to managing software development projects that focuses on continuous releases and incorporating customer feedback with every iteration.



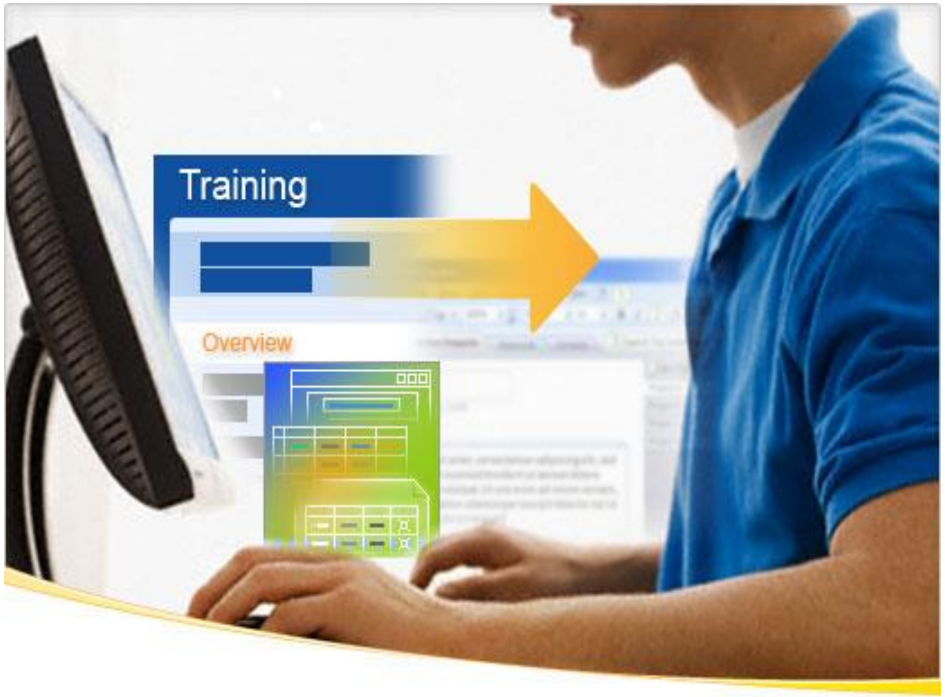
APM- Why

- Increased Consumer Expectations:
  - **As consumers today** we want and expect innovative products: *faster*, *cheaper* and with *better quality* than those we’ve seen in the past.

Consumer Expectations

APM- Why

- Increased Work Pressure:
  - **As knowledge workers**, our business tools have improved our *capability to be productive*, raising work expectations.



Why APM

- Traditional Project Management Practices *can* Lead to:
  - Chaos – Junior Project Managers tend to either:
    - allow too much uncontrolled changed to take place (to ensure customer satisfaction) or
    - are too strict in allowing for change (resulting in irate customers).
  - Dramatic Project Underperformance – According to the Standish Group's *Chaos Reports*, only 25 percent of IT projects are successful, the remainder are:
    - Late.
    - Over Budget.
    - Deliver only a fraction of original scope in order to meet budget restrictions.
    - Cancelled.

Traditional PM versus Agile Methods

- Traditional PM Approach
  - Concentrates on thorough, upfront planning of the entire project.
  - Requires a high degree of predictability to be effective.
- Agile Project Management (Agile PM)
  - Relies on incremental, iterative development cycles to complete less-predictable projects.
  - Is ideal for exploratory projects in which requirements need to be discovered and new technology tested.
  - Focuses on active collaboration between the project team and customer representatives.

Traditional PM versus Agile Methods

Traditional	**Agile**
Design up front	Continuous design
Fixed scope	Flexible
Deliverables	Features/requirements
Freeze design as early as possible	Freeze design as late as possible
Low uncertainty	High uncertainty
Avoid change	Embrace change
Low customer interaction	High customer interaction
Conventional project teams	Self-organized project teams

Change in focus

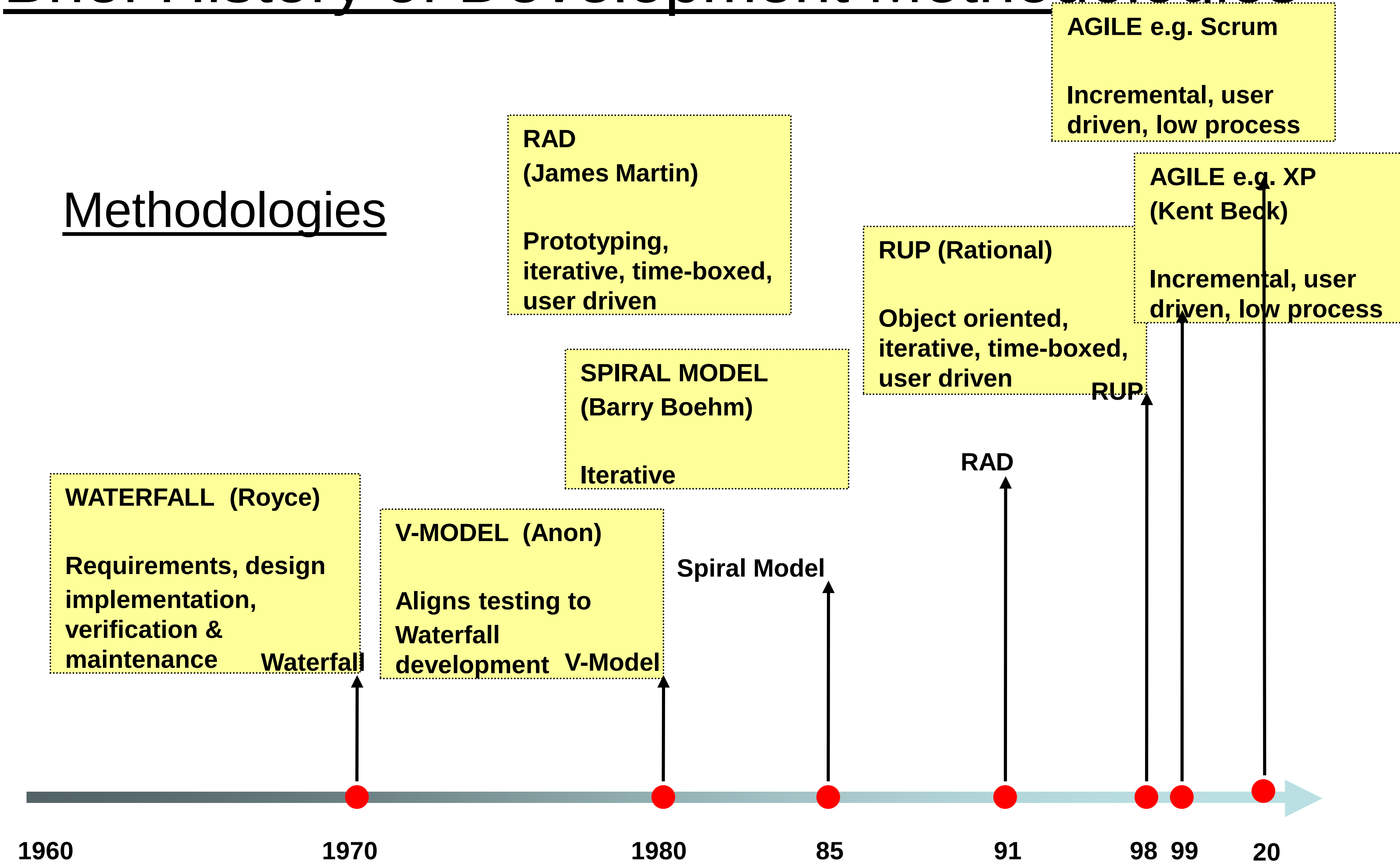
- Traditional PM focus on requirements to set the scope, and then concentrates on delivering those requirements
- In Agile we focus on delivering value and are constantly questioning the scope

What is different about Agile Methods?

- Short iterations:
  - used to keep the feedback flowing (allowing for increased responsiveness to change and reducing the risk of building the wrong thing).
- Open, Flexible and Extensive:
  - design using open standards whenever possible
- Empowered Teams:
  - Experienced specialists are encouraged to work out the detail design on their own.
- Personal Communication:
  - Rather than relying on written documentation to communicate design decisions, technical approaches and other typically documented items, agile method suggest that the team work in the same physical space (co-location).
    - Use of white boards in the work area is encouraged rather than lengthy formal detail design documentation.

Brief History of Development Methodologies

Methodologies



WATERFALL (Royce)

Requirements, design implementation, verification & maintenance

V-MODEL (Anon)

Aligns testing to Waterfall development

SPIRAL MODEL (Barry Boehm)

Iterative

RAD (James Martin)

Prototyping, iterative, time-boxed, user driven

RUP (Rational)

Object oriented, iterative, time-boxed, user driven

AGILE e.g. XP (Kent Beck)

Incremental, user driven, low process

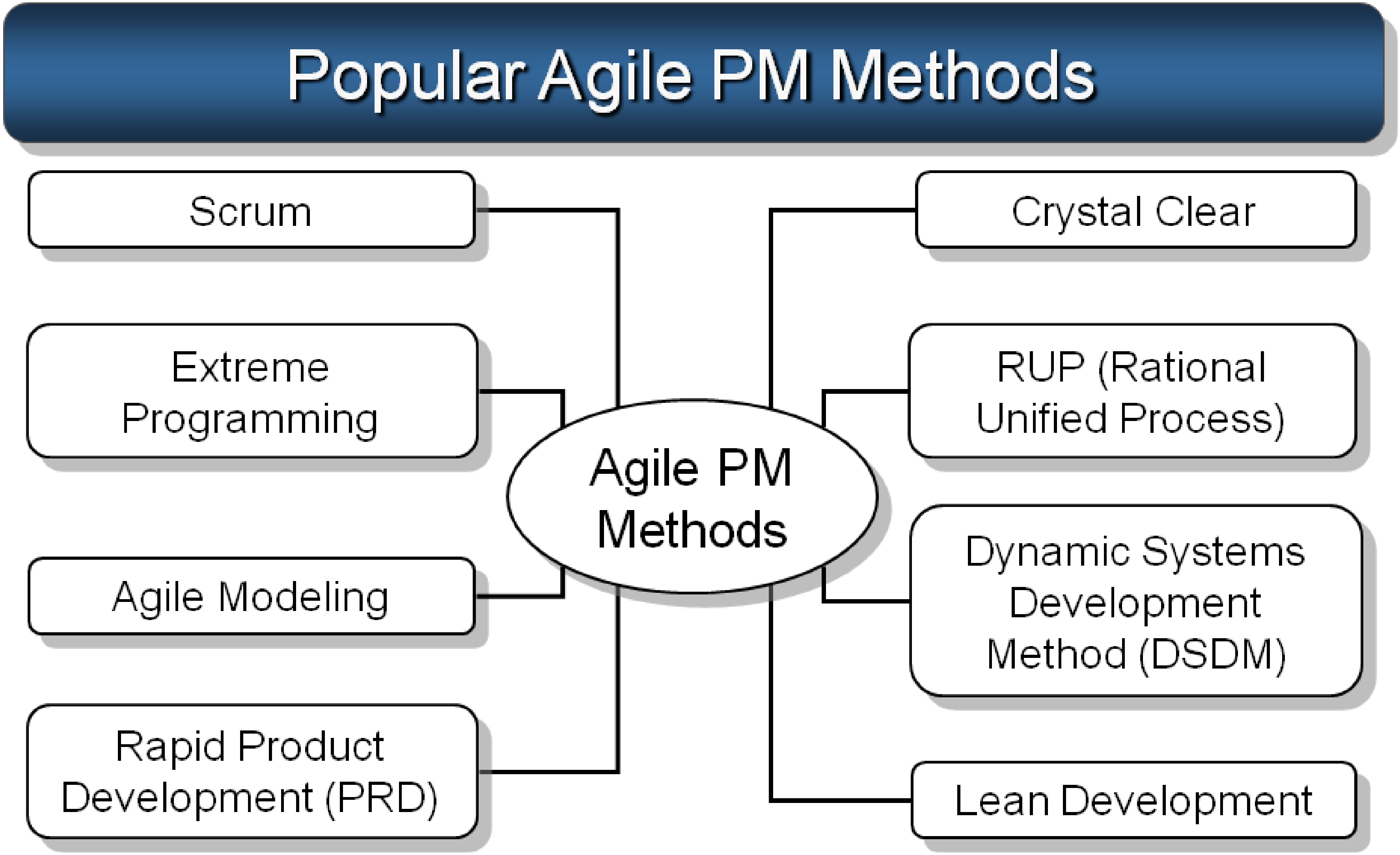
AGILE e.g. Scrum

Incremental, user driven, low process

The Benefits of Being Agile

- **Reducing Risk** – The benefits from improved control and improved communication lead to reduced risks. Examples of risks include:
  - **Risk of building (or doing) the wrong thing.**
    - Did the sponsor get what they asked for but not what they actually wanted?
  - **Risk of building the right thing poorly.**
    - For example, was the product poorly crafted. Was it thoroughly tested as a part of each iteration? Is the final produce extensible?
  - **Risk of being placed into an endless cycle of design updates and reviews**
    - due to changing requirements or high levels of complexity

The Agile Landscape



Popular Agile PM Methods

Agile PM Methods

Scrum

Extreme Programming

Agile Modeling

Rapid Product Development (PRD)

Crystal Clear

RUP (Rational Unified Process)

Dynamic Systems Development Method (DSDM)

Lean Development

The Agile Landscape

Agile PM Principles

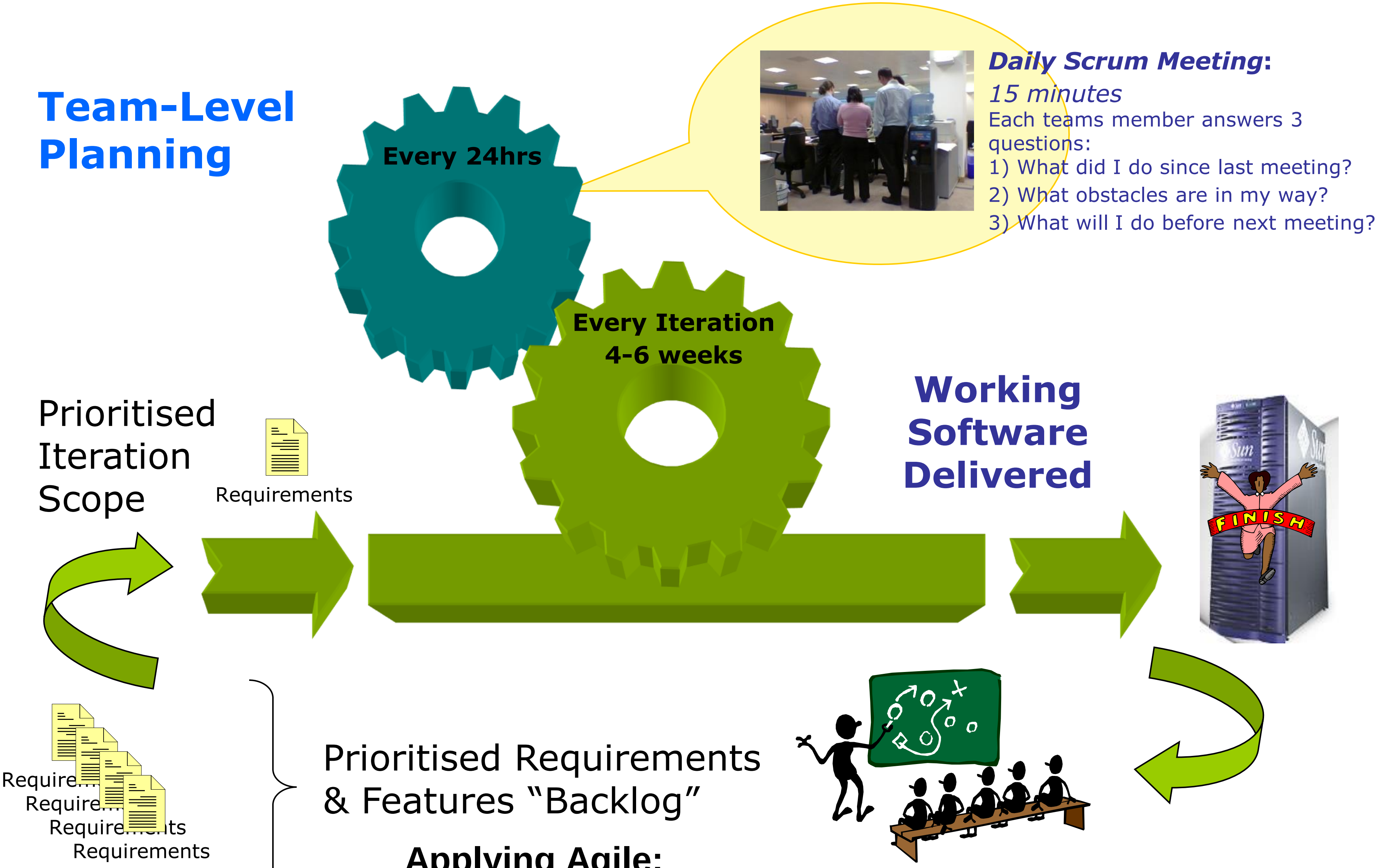
- Focus on customer value
- Iterative and incremental delivery
- Experimentation and adaptation
- Self-organization
- Continuous improvement

Scrum

- Definition from rugby football:
  - A scrum is a way to restart the game after an interruption, where the forwards of each side come together in a tight formation and struggle to gain possession of the ball when it is tossed in among them
- SCRUM is an agile, lightweight process for managing and controlling software and product development in rapidly changing environments.

- Iterative, incremental process
- Team-based approach
- developing systems/ products with rapidly changing requirements
- Controls the chaos of conflicting interest and needs
- Improve communication and maximize cooperation
- Protecting the team form disruptions and impediments
- A way to maximize productivity

Agile project management - SCRUM



Team-Level Planning

Every 24hrs

***Daily Scrum Meeting*:**

15 minutes

Each teams member answers 3 questions:

1) What did I do since last meeting?

2) What obstacles are in my way?

3) What will I do before next meeting?

Every Iteration
4-6 weeks

Prioritised Iteration Scope

Requirements

Working Software Delivered

Require
Require
Require ts
Requirements

Prioritised Requirements & Features "Backlog"

Applying Agile:
Continuous integration; continuously monitored progress

Scrum Master

- Represents management to the project
- Typically filled by a Project Manager or Team Leader
- Responsible for enacting scrum values and practices
- Main job is to remove impediments

Scrum Team

- Typically 5-10 people
- Cross-functional (QA, Programmers, UI Designers, etc.)
- Members should be full-time
- Team is self-organizing
- Membership can change only between sprints

Product Owner

- Acts like one voice (in any case)
- Knows what needs to be build and in what sequence this should be done
- Typically a product manager

Scrum Process

- Sprint Planning Meeting
  - A collaborative meeting in the beginning of each Sprint between the Product Owner, the Scrum Master and the Team
    - Takes 8 hours and consists of 2 parts (“before lunch and after lunch”)
- 1st Part:
  - Creating Product Backlog
  - Determining the Sprint Goal.
  - Participants: Product Owner, Scrum Master, Scrum Team
- 2nd Part:
  - Participants: Scrum Master, Scrum Team
  - Creating Sprint Backlog

Pre-Project/Kickoff Meeting

- A special form of Sprint Planning Meeting
- Meeting before the begin of the Project

Scrum Process

- Sprint
  - A month-long iteration, during which is incremented a product functionality
  - NO outside influence can interference with the Scrum team during the Sprint
  - Each Sprint begins with the Daily Scrum Meeting
- Daily Scrum
  - Is a short (15 minutes long) meeting, which is held every day before the Team starts working
  - Participants: Scrum Master (which is the chairman), Scrum Team
  - Every Team member should answer on 3 questions

Questions

- What did you do since the last Scrum?
- What are you doing until the next Scrum?
- What is stopping you getting on with the work?

Daily Scrum

- Is NOT a problem solving session
- Is NOT a way to collect information about WHO is behind the schedule
- Is a meeting in which team members make commitments to each other and to the Scrum Master
- Is a good way for a Scrum Master to track the progress of the Team

Scrum Process

- Sprint Review Meeting
  - Is held at the end of each Sprint
  - Business functionality which was created during the Sprint is demonstrated to the Product Owner
  - Informal, should not distract Team members of doing their work

- Please Visit
  - http://scrumtrainingseries.com/BacklogRefinementMeeting/BacklogRefinementMeeting.htm

Scrum Artifacts

- Product Backlog
  - Requirements for a system, expressed as a prioritized list of Backlog Items
  - Is managed and owned by a Product Owner
  - Spreadsheet (typically)
  - Usually is created during the Sprint Planning Meeting
  - Can be changed and re-prioritized before each PM
- Is only a FORECAST!-> is not exact

Estimation of Product Backlog Items

- Establishes team’s velocity (how much Effort a Team can handle in one Sprint)
- Determining units of complexity.
  - Size-category (“T-Shirt size”)
  - Story points
  - Work days/work hours
- Methods of estimation:
  - Expert Review
  - Creating a Work Breakdown Structure (WBS)

Scrum Artifacts

- Sprint Backlog
  - A subset of Product Backlog Items, which define the work for a Sprint
  - Is created ONLY by Team members
  - Each Item has it’s own status
  - Should be updated every day
- Is a FORECAST!
- Is a good warning monitor

Sprint Backlog

- No more then 300 tasks in the list
- If a task requires more than 16 hours, it should be broken down
- Team can add or subtract items from the list. Product Owner is not allowed to do it

Scrum Artifacts

- Burn down Charts
  - Are used to represent “work done”.
  - Are wonderful Information Radiators
  - 3 Types:
    - Sprint Burn down Chart (progress of the Sprint)
    - Release Burn down Chart (progress of release)
    - Product Burn down chart (progress of the Product)
- Please visit
  - http://scrummethodology.com/scrum-effort-estimation-and-story-points/

Referneces

- Abrahamsson P, Salo O and Ronkainen J. Agile software development methods (Review and analysis).
- Scott W Ambler. Agile model driven development.
- Cohen D, Lindvall M, Costa P. Agile software development.
- http://www.agilemanifesto.org/
- http://www.scrum-master.com/
- Agile Software Development Tutorial: How to Transition to Agile: http://searchsoftwarequality.techtarget.com/tutorial/Agile-software-development-tutorial-How-to-transition-to-agile

Acquiring the Project Team

- Acquiring qualified people for teams is crucial
  - The project manager who is the smartest person on the team has done a poor job of recruiting!
- Staffing plans and good hiring procedures are important, as are incentives for recruiting and retention
  - Some companies give their employees one dollar for every hour a new person they helped hire works
  - Some organizations allow people to work from home as an incentive

Developing the Project Team

- The main goal of **team development** is to help people work together more effectively to improve project performance
  - It takes teamwork to successfully complete most projects
- Tuckman Model of Team Development
  - Forming
  - Storming
  - Norming
  - Performing
  - Adjourning

The approaches just mentioned introduce some flexibility while retaining the waterfall’s main characteristics. In many situations, however, even these changes are not sufficient and a more radical approach is necessary

1. **Allowing for structural backtracking during the testing phase (V-cycle model)**. In the V-cycle model, testing activities provide systematic ways to consolidate the implementation or backtrack to the most appropriate development activity
2. **Allowing for an iterative and evolutionary development of the system (The Spiral).** All the phases are repeated various times to deliver increasingly refined versions of the system.
3. **Blurring the boundaries between activities. Prototyping/Evolutionary**

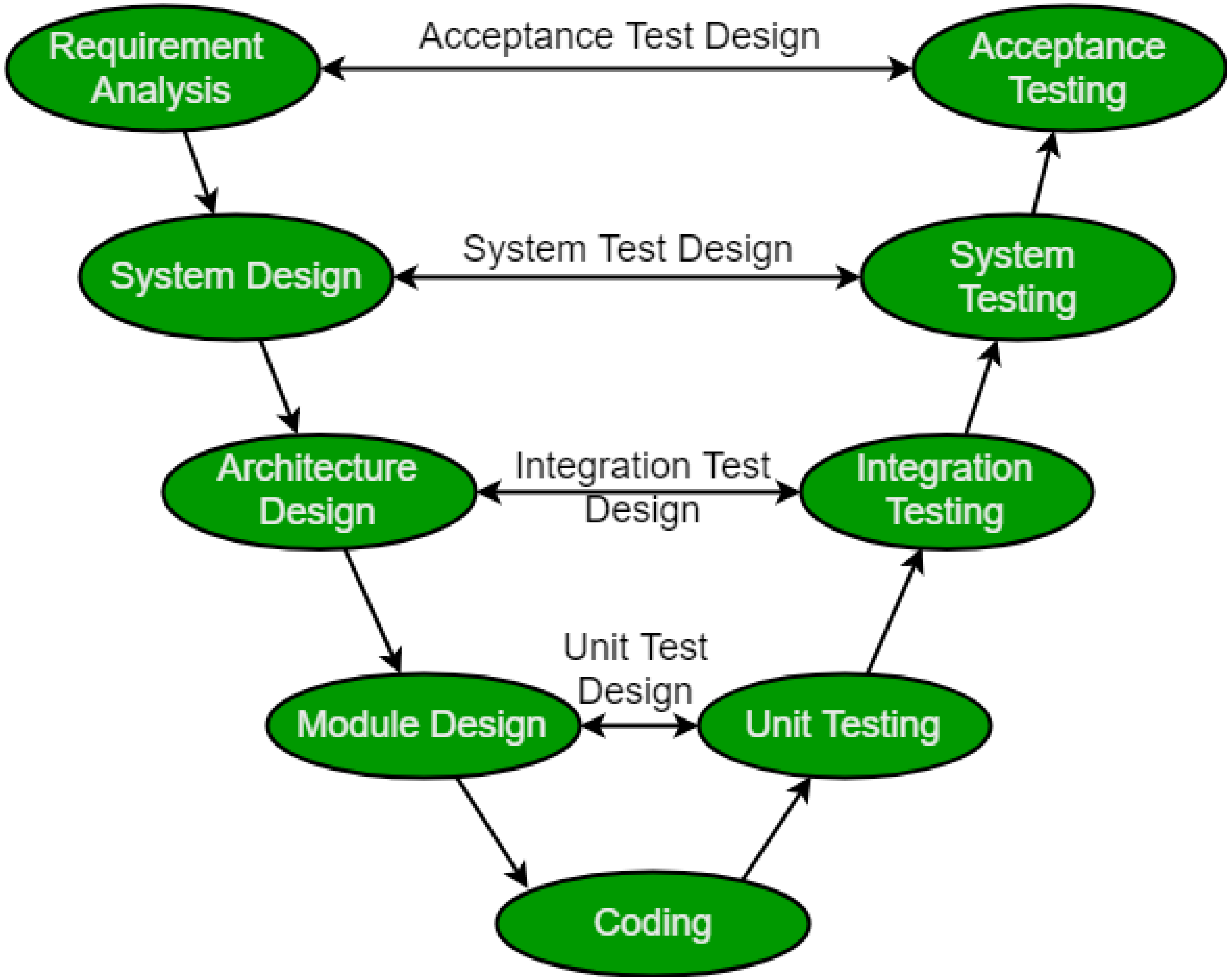
V-model (Agile V Model)

The V-model is a type of SDLC model where process executes in a sequential manner in V-shape. It is also known as Verification and Validation model. It is based on the association of a testing phase for each corresponding development stage. Development of each step directly associated with the testing phase. The next phase starts only after completion of the previous phase i.e. for each development activity, there is a testing activity corresponding to it.

Verification and Validation phases are joined by coding phase in V-shape. Thus it is called V-Model.

Verification: It involves static analysis technique (review) done without executing code. It is the process of evaluation of the product development phase to find whether specified requirements meet.

Validation: It involves dynamic analysis technique (functional, non-functional), testing done by executing code. Validation is the process to evaluate the software after the completion of the development phase to determine whether software meets the customer expectations and requirements.


Requirement Analysis
Acceptance Test Design
Acceptance Testing
System Design
System Test Design
System Testing
Architecture Design
Integration Test Design
Integration Testing
Unit Test Design
Module Design
Unit Testing
Coding