AGILE methodology- Scrum

What is Agile?

This is one of the biggest *buzzwords* in the IT industry these days. But, what exactly is agile?

The Agile model provides alternatives to traditional project management. Agile approaches are used in software development to respond to unpredictability. In this tutorial we will learn in brief about agile and related concepts.

To simply define it's a new way of managing projects and teams in software industry.

Agile Manifesto

Group of people formed agile manifesto uncovering ways of developing software. Which says that they value:

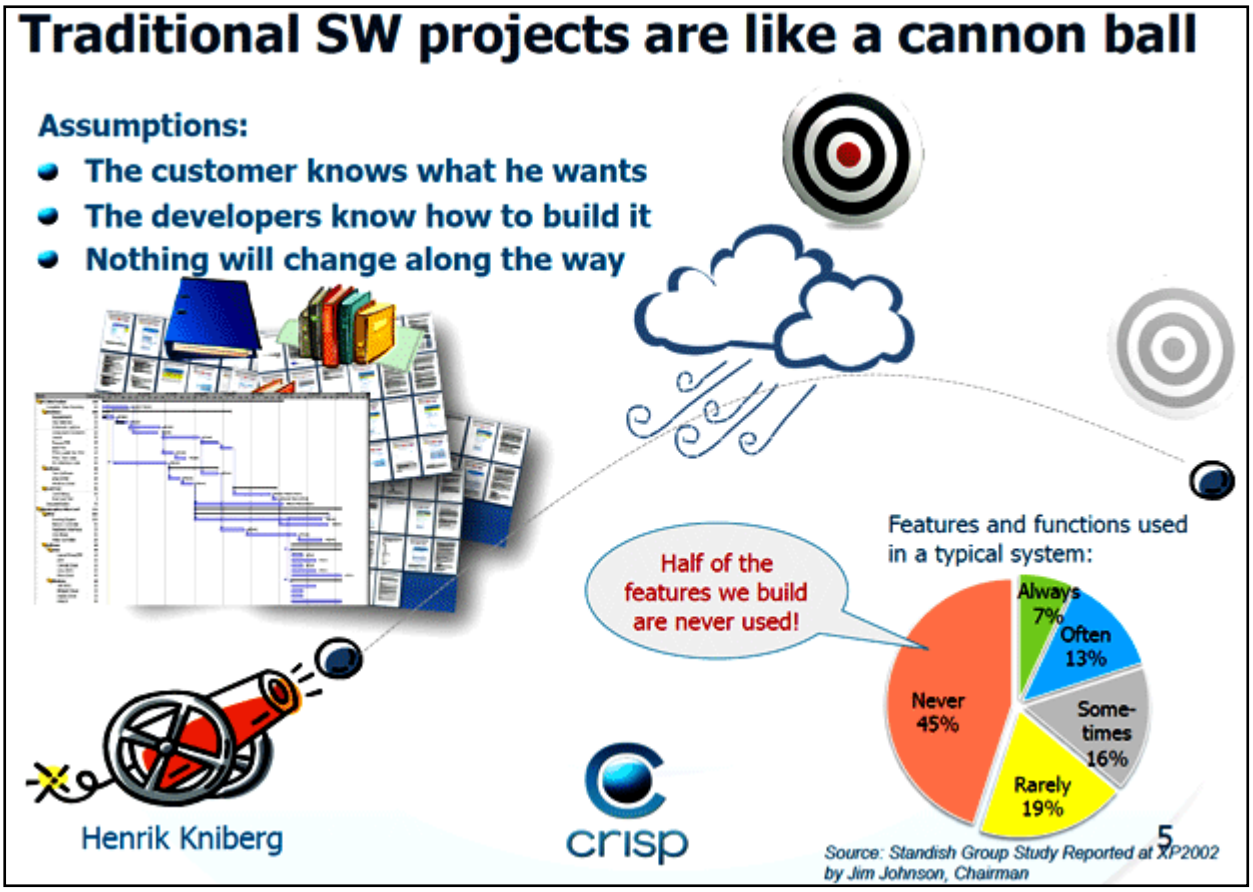
Individuals and interactions over processes and tools

Working software over comprehensive documentation

Customer collaboration over contract negotiation

Responding to change over following a plan

- Agile development means you're doing small cycles of the complete SDLC for a project.
- Each cycle lasts a certain number of weeks and you do the planning, requirements analysis, design, coding, and testing in that cycle.
- Then move on to the the next cycle and do the same thing until you get the project completed.
- This process allows for less documentation up front, gathering requirements as you go along, and requires heavier involvement from the business throughout the project.
- Agile promotes working together – typically in a large office space with everyone from the team together – and relies more on face-to-face discussion than documentation.



12 key Principles in Agile:

Priority: To satisfy the customer through early and continuous delivery of valuable software.

Accept Change: Welcome changes, even late in development.

Work Together: Business people and developers must work together daily throughout the project.

Trust Individuals: Build projects around motivated individuals, give them the environment required and trust them for getting the job completed.

Face to Face Conversion: The effective way for communication within the team is face to face conversion.

Progress: working software is the effective measure of progress.

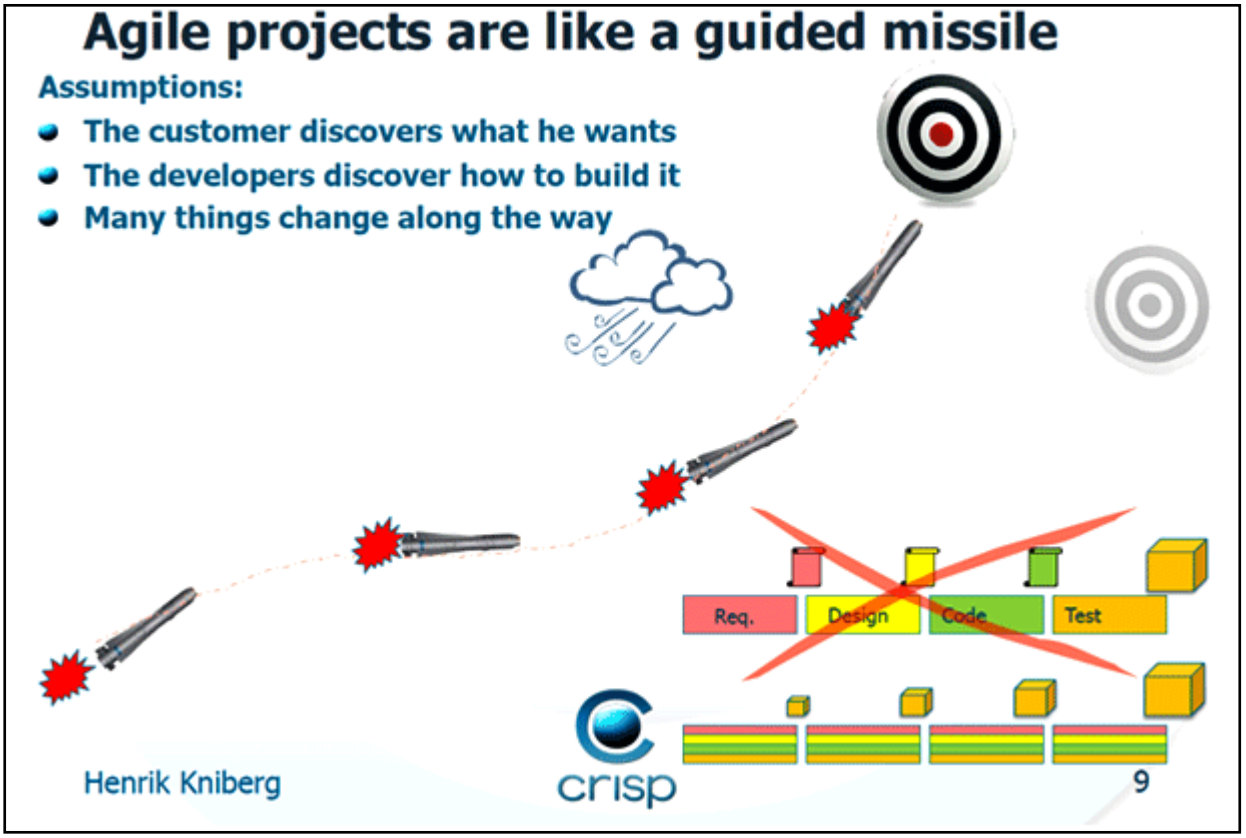
Maintain Pace: All the stakeholders should maintain a constant pace as it offers sustainable development.

Technicalities and Design: Continuous attention to technical excellence and good design enhances agility.

Simplicity: The art of maximizing the amount of work not done–is essential.

Self Organized Teams: The best architectures, requirements, and designs emerge from self-organizing teams.

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



SCRUM:

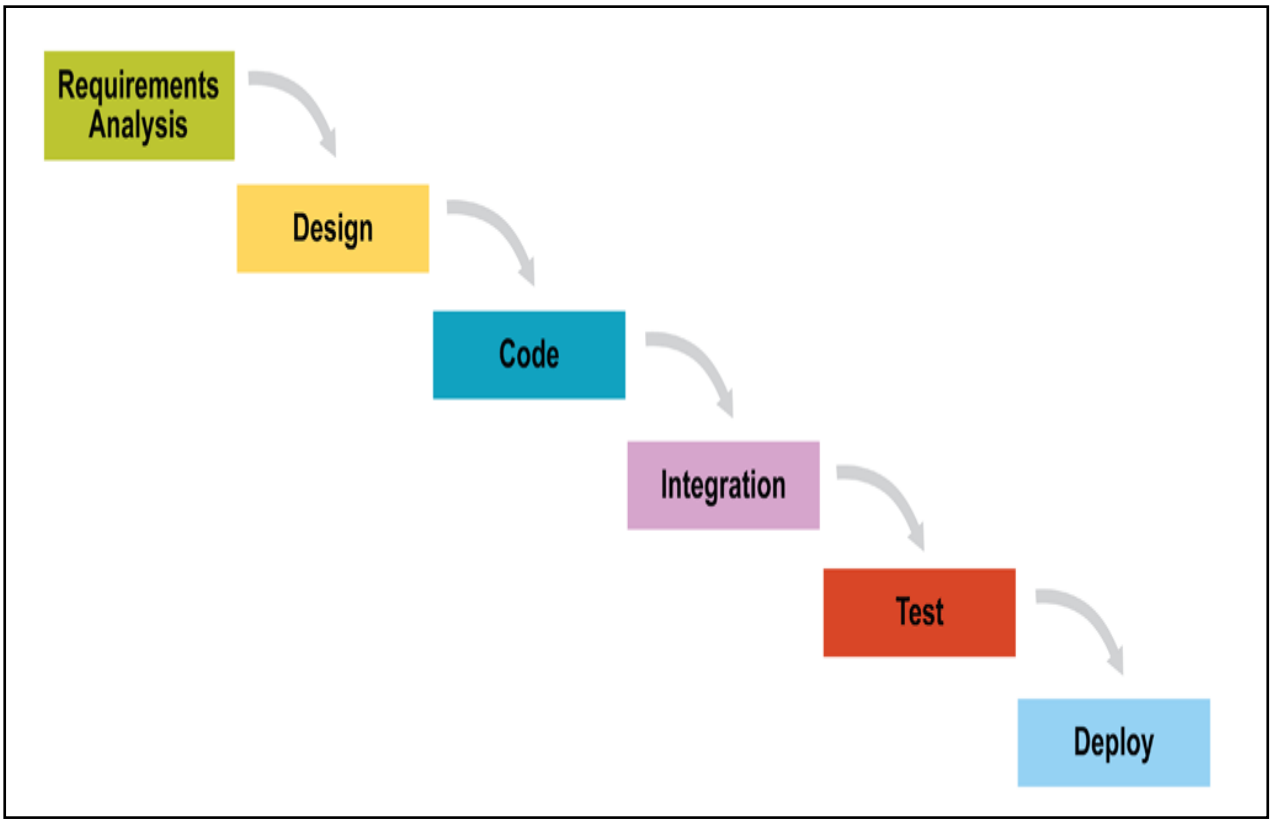
Scrum is a management framework for incremental product development using one or more cross-functional, self-organizing teams of about seven people each.

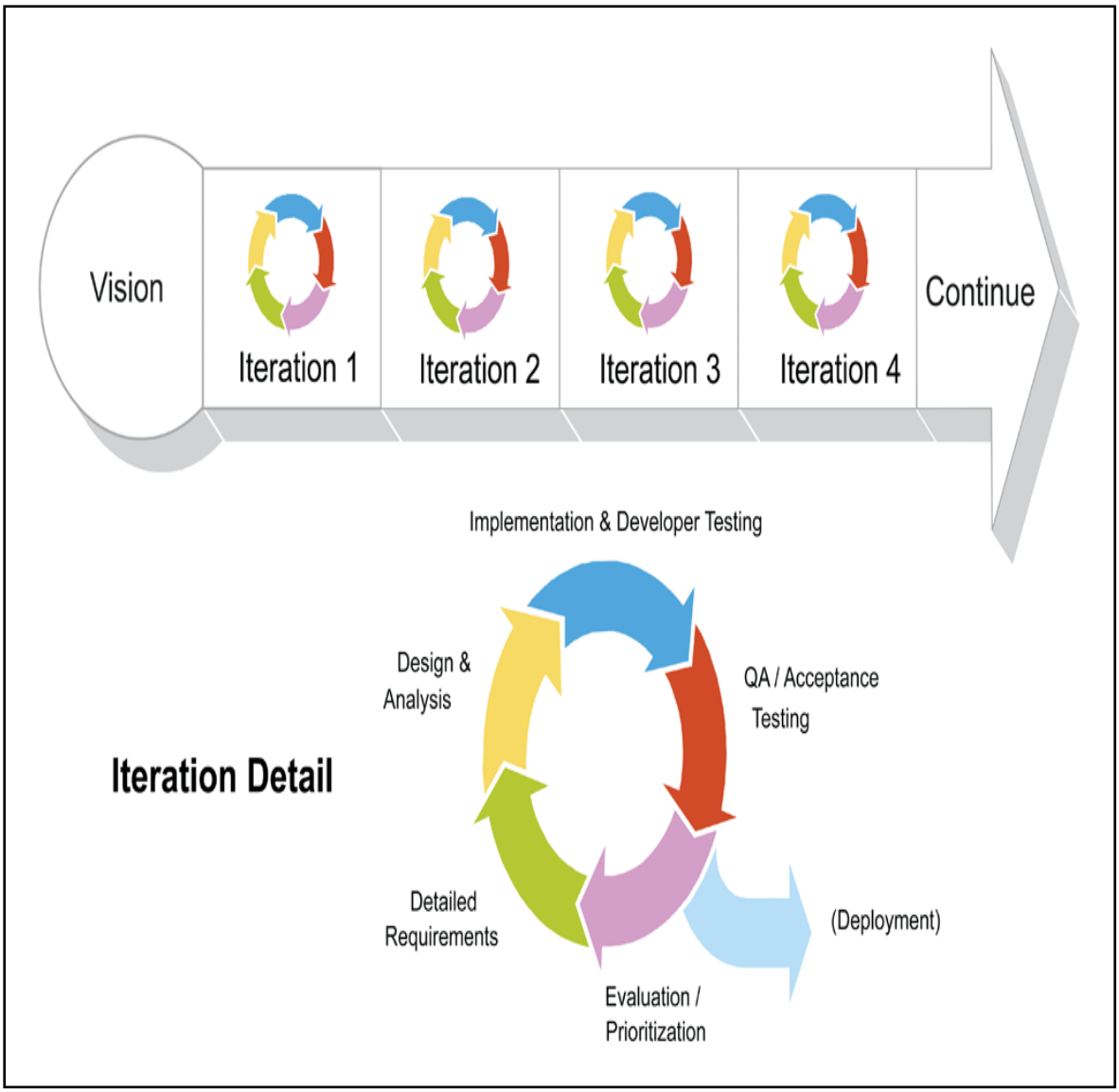
It provides a structure of roles, meetings, rules, and artifacts. Teams are responsible for creating and adapting their processes within this framework.

Scrum uses fixed-length iterations, called Sprints, which are typically 1-2 weeks long (never more than 30 days). Scrum teams attempt to build a potentially shippable (properly tested) product increment every iteration.

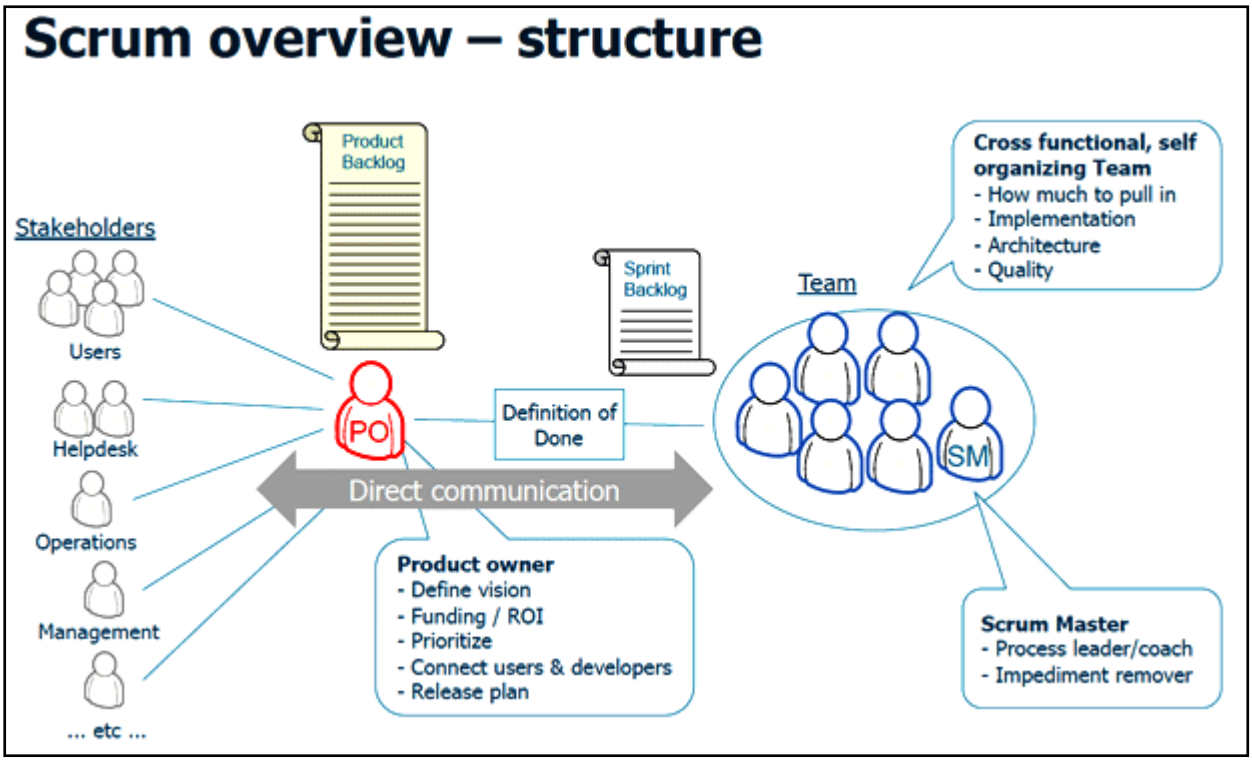
An Alternative to Waterfall:

Scrum's incremental, iterative approach trades the traditional phases of "waterfall" development for the ability to develop a subset of high-value features first, incorporating feedback sooner.





The greatest potential benefit of Scrum is for complex work involving knowledge creation and collaboration, such as new product development. Scrum is usually associated with object-oriented software development. Its use has also spread to the development of products such as semiconductors, mortgages, and wheelchairs.



Doing Scrum, or Pretending to Do Scrum?

Scrum's relentless reality checks expose dysfunctional constraints in individuals, teams, and organizations. Many people claiming to do Scrum modify the parts that require breaking through organizational impediments and end up robbing themselves of most of the benefits.

Scrum Roles:

Product Owner

- Single person responsible for maximizing the return on investment (ROI) of the development effort
- Responsible for product vision
- Constantly re-prioritizes the Product Backlog, adjusting any longterm expectations such as release plans
- Final arbiter of requirements questions
- Accepts or rejects each product increment

- Decides whether to ship
- Decides whether to continue development
- Considers stakeholder interests
- May contribute as a team member

Scrum Development Team

- Cross-functional (e.g., includes members with testing skills, and often others not traditionally called developers: business analysts, domain experts, etc.) Self-organizing / self-managing, without externally assigned roles
- Negotiates commitments with the Product Owner, one Sprint at a time
- Has autonomy regarding how to reach commitments
- Intensely collaborative
- Most successful when located in one team room, particularly for the first few Sprints
- Most successful with long-term, full-time membership. Scrum moves work to a flexible learning team and avoids moving people or splitting them between teams.
- 7 ± 2 members

ScrumMaster

- Facilitates the Scrum process
- Helps resolve impediments
- Creates an environment conducive to team self-organization
- Captures empirical data to adjust forecasts
- Shields the team from external interference and distractions to keep it in group flow (a.k.a. the zone)
- Enforces timeboxes
- Keeps Scrum artifacts visible
- Promotes improved engineering practices

- Has no management authority over the team (anyone with authority over the team is by definition not its ScrumMaster)

Scrum Meetings