Marwadi University

Presented By,

Mr. Sanjay Togadiya
Founder, RK Infotech, Rajkot

Mr. Siddharth Jogia
Project Manager, RK Infotech, Rajkot

www.rkinfotechindia.com

RK INFOTECH
Unwinding IT Solution

About RK Infotech

RK INFOTECH is a leading IT company in Rajkot, India empowering business transformation for enterprises and technology providers by delivering customized, reliable and cost efficient solutions.

We believe in quality and always thrive to provide state of the art technology in all the relative fields.

- 5+ years experience IT company
- 27+ experts teams
- Global Presence
- 400+ Domestic + International satisfied clients

Our Mission

To be the best global IT Solution Company by providing exceptional cost effective solutions and services to our clients, surpassing their expectations and enhancing revenues at the same time.

Head Office

SF-48, Shreemad Bhavan, Opp. Kanta
Vikas Gruh Road, Bhaktinagar Main
Road, Rajkot 360002 (Gujarat, India)

Branch Office

B-4/50C, Ashok Vihar,Phase-2,
Near Sagar Ratna, New Delhi -
110052 (India)

Sister Company

ConductExam™
Create, Share & Analyze

PIXIBIT
Design Studio

Our Products & Services

Software Products

- Online Exam Software
- Business ERP Software
- Accounting Software
- Question Paper Generator
- USB Based Study Software
- Bulk Cheque Printing Software
- School College ERP System
- Label / Barcode Printing Software
- Store Management Software

Customized Solutions

- e-Commerce Website
- CMS Website
- Business ERP
- Financial Solution
- CRM System

Working as IT Partners

- IT Partners
- IOTs
- KPO

Other Services & Connected Industries

Other Services

- Website Development
- Android App Development
- iOS App Development
- Software Development
- Search Engine Optimization
- Graphics Designs
- Customized ERP Solutions
- Customized CRM Solutions

Industries We Serve

- Education & e- Learning
- Automobile Industry
- SME Industries
- Financial Services
- Real Estate
- Hospitality & Healthcare
- Logistic Services
- Information Services
- Travel & Tourism
- Retail & e-Commerce Industry

Introduction to Agile

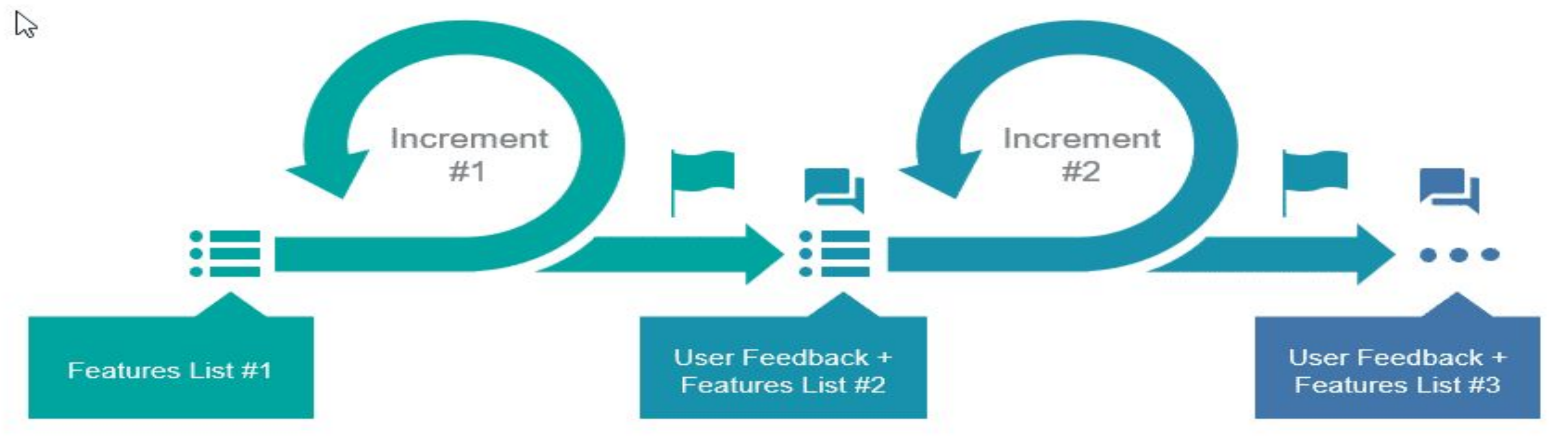
Agile History

In 2001, this new management paradigm began to pick up momentum, agile was formalized when 17 pioneers of the agile methodology met at the Snowbird Ski Resort in Utah and issued the Agile Manifesto.

1. Kent Beck
2. Mike Beedle
3. Arie van Bennekum
4. Alistair Cockburn
5. Ward Cunningham
6. Martin Fowler
7. James Grenning
8. Jim Highsmith
9. Andrew Hunt
10. Ron Jeffries
11. Jon Kern
12. Brian Marick
13. Robert C. Martin
14. Steve Mellor
15. Ken Schwaber
16. Jeff Sutherland
17. Dave Thomas

What is Agile?

Agile is a time boxed, iterative approach to software delivery that builds software incrementally from the start of the project, instead of trying to deliver it all at once near the end. It works by breaking projects down into little bits of user functionality called user stories, prioritizing them, and then continuously delivering them in short two week cycles called iterations.


Increment #1
Increment #2
Features List #1
User Feedback + Features List #2
User Feedback + Features List #3

How does it work?

1. **You make a list**

Sitting down with your customer you make a list of features they would like to see in their software. We call these things user stories and they become the To Do list for your project.

2. **You size things up**

You size(estimate) your stories relatively to each other, coming up with a guess as to how long you think each user story will take.

3. **You set some priorities**

Like most lists, there always seems to be more to do than time allows. So you ask your customer to prioritize their list so you get the most important stuff done first, and save the least important for last.

How does it work? [Continue...]

4. **You start executing**

Then you start delivering some value. You start at the top. Work your way to the bottom. Building, iterating, and getting feedback from your customer as you go.

5. **You update the plan as you go**

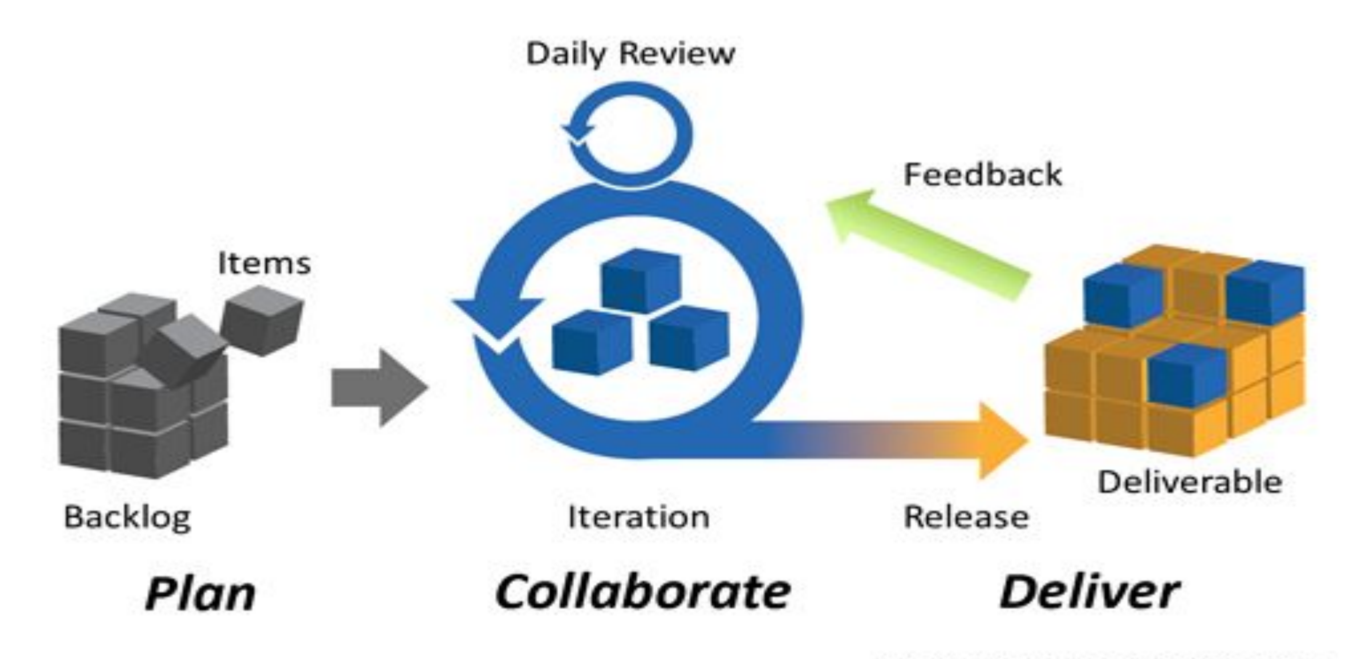
Then as you and your customer starting delivering one of two things is going to happen. You'll discover:

- You're going fast enough. All is good. Or,
- You have too much to do and not enough time.

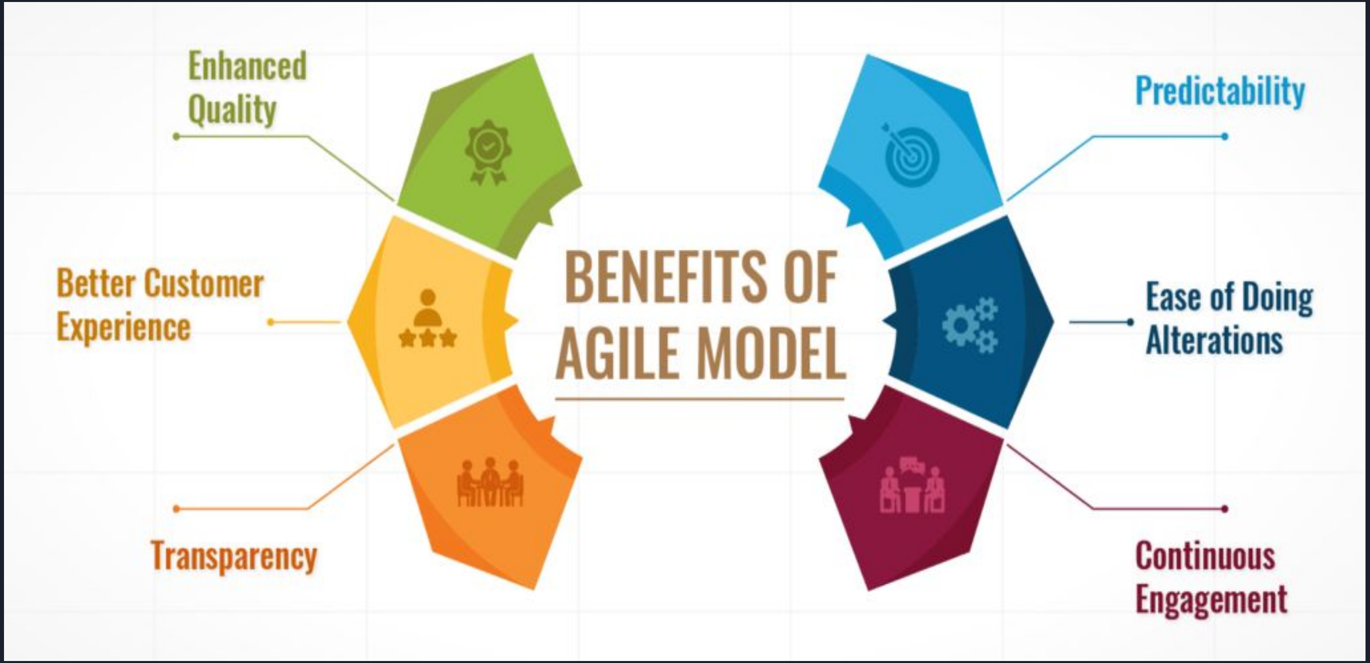
At this point you have two choices. You can either

A. do less and cut scope (recommended). Or
B. you can push out the date.

How does it work? [Continue...]



Why Use Agile Methods



Advantages of Agile Model

- **Customer satisfaction by rapid, continuous delivery of useful software.**
- **People and interactions are emphasized rather than process and tools. Customers, developers and testers constantly interact with each other.**
- **Working software is delivered frequently (weeks rather than months).**
- **Face-to-face conversation is the best form of communication.**
- **Close, daily cooperation between business people and developers.**
- **Continuous attention to technical excellence and good design.**
- **Regular adaptation to changing circumstances.**
- **Even late changes in requirements are welcomed**

Disadvantages of Agile Model

- **In case of some software deliverables, especially the large ones, it is difficult to assess the effort required at the beginning of the software development life cycle.**

- **The project can easily get taken off track if the customer representative is not clear what final outcome that they want.**

- **Only senior programmers are capable of taking the kind of decisions required during the development process. Hence it has no place for newbie programmers, unless combined with experienced resources.**

THE AGILE MANIFESTO

We are uncovering better ways of developing software by doing it and helping others do it.
Through this work we have come to 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

The Twelve Agile Manifesto Principles

1. **Customer satisfaction through early and continuous software delivery -** Customers are happier when they receive working software at regular intervals, rather than waiting extended periods of time between releases.

2. **Accommodate changing requirements throughout the development process -** The ability to avoid delays when a requirement or feature request changes.

3. **Frequent delivery of working software –** Scrum accommodates this principle since the team operates in software sprints or iterations that ensure regular delivery of working software.

The Twelve Agile Principles [Continue...]

4. **Collaboration between the business stakeholders and developers throughout the project –** Better decisions are made when the business and technical team are aligned.

5. **Support, trust, and motivate the people involved –** Motivated teams are more likely to deliver their best work than unhappy teams.

6. **Enable face-to-face interactions –** Communication is more successful when development teams are co-located.

The Twelve Agile Principles [Continue...]

7. **Working software is the primary measure of progress –** Delivering functional software to the customer is the ultimate factor that measures progress.

8. **Agile processes to support a consistent development pace –** Teams establish a repeatable and maintainable speed at which they can deliver working software, and they repeat it with each release.

9. **Attention to technical detail and design enhances agility –** The right skills and good design ensures the team can maintain the pace, constantly improve the product, and sustain change.

10. **Simplicity –** Develop just enough to get the job done for right now.

The Twelve Agile Principles [Continue...]

11. **Self-organizing teams encourage great architectures, requirements, and designs –** Skilled and motivated team members who have decision-making power, take ownership, communicate regularly with other team members, and share ideas that deliver quality products.

12. **Regular reflections on how to become more effective –** Self-improvement, process improvement, advancing skills, and techniques help team members work more efficiently.

The intention of Agile is to align development with business needs, and the success of Agile is apparent.

Agile projects are customer focused and encourage customer guidance and participation. As a result, Agile has grown to be an overarching view of software development throughout the software industry and an industry all by itself.

The Twelve Agile Principles [Continue...]

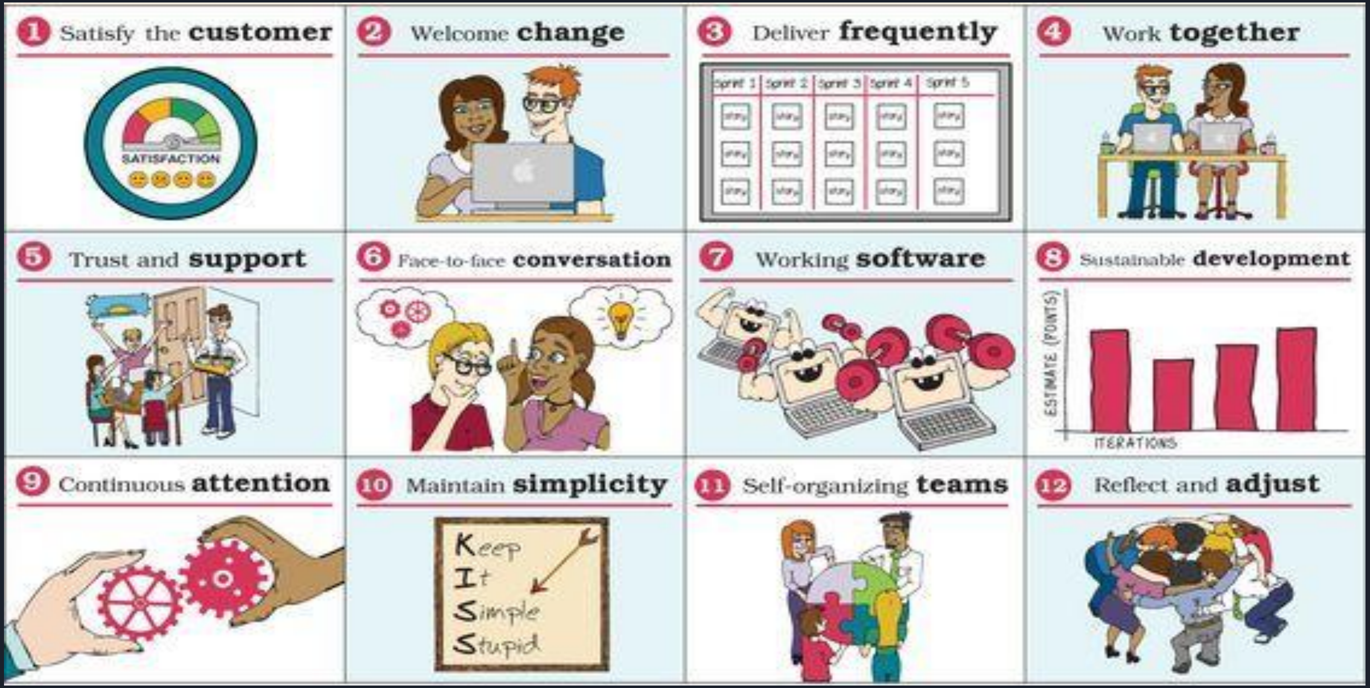
11. **Self-organizing teams encourage great architectures, requirements, and designs –** Skilled and motivated team members who have decision-making power, take ownership, communicate regularly with other team members, and share ideas that deliver quality products.

12. **Regular reflections on how to become more effective –** Self-improvement, process improvement, advancing skills, and techniques help team members work more efficiently.

The intention of Agile is to align development with business needs, and the success of Agile is apparent.

Agile projects are customer focused and encourage customer guidance and participation. As a result, Agile has grown to be an overarching view of software development throughout the software industry and an industry all by itself.

The Twelve Agile Principles [Continue...]



Agile Methods & Practices

What are Agile Methods?

Is this what we mean by agile method:

```
Class Agile
{
    public function agileMethods () {
            getMethod(Agile.agile);
    }
}
```

Definition by Agile Philosophy

Agile methods are processes that support the agile philosophy. Examples include Extreme Programming and Scrum, FDD, TDD, etc.

Agile methods consist of individual elements called practices.

Practices include using version control, setting coding standards, and giving weekly demos to your stakeholders.

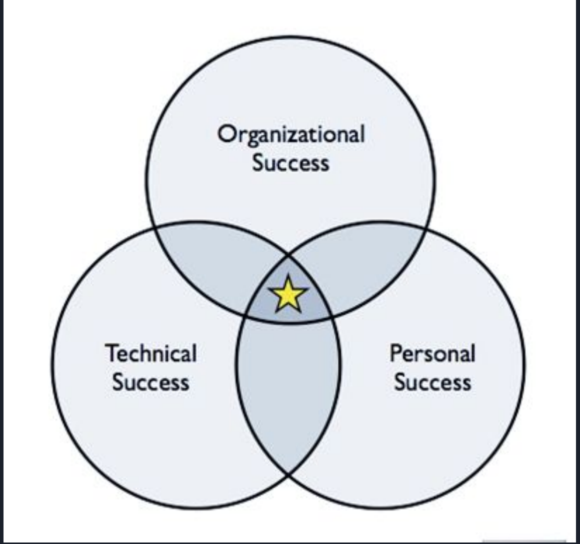
Agile Methods combine these practices to accentuate parts that support agile philosophy.

Aspect of Success with Agile Methods

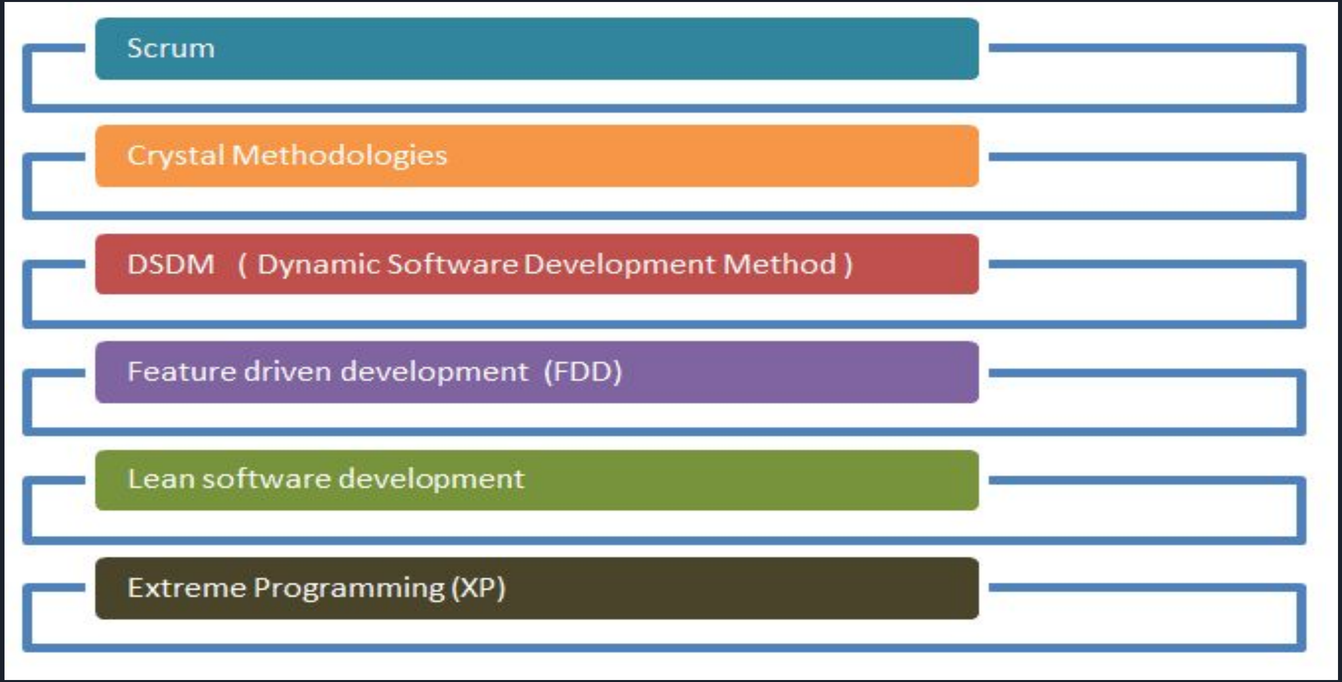
- Organization success is about delivering value – and agile philosophy has this at the heart of it.

- Personal success is, well Personal, but everyone gains from improved collaboration.

- Technical success, because of practices like TDD, Scrum and extreme programming you are continuously improving. This overlaps with personal.

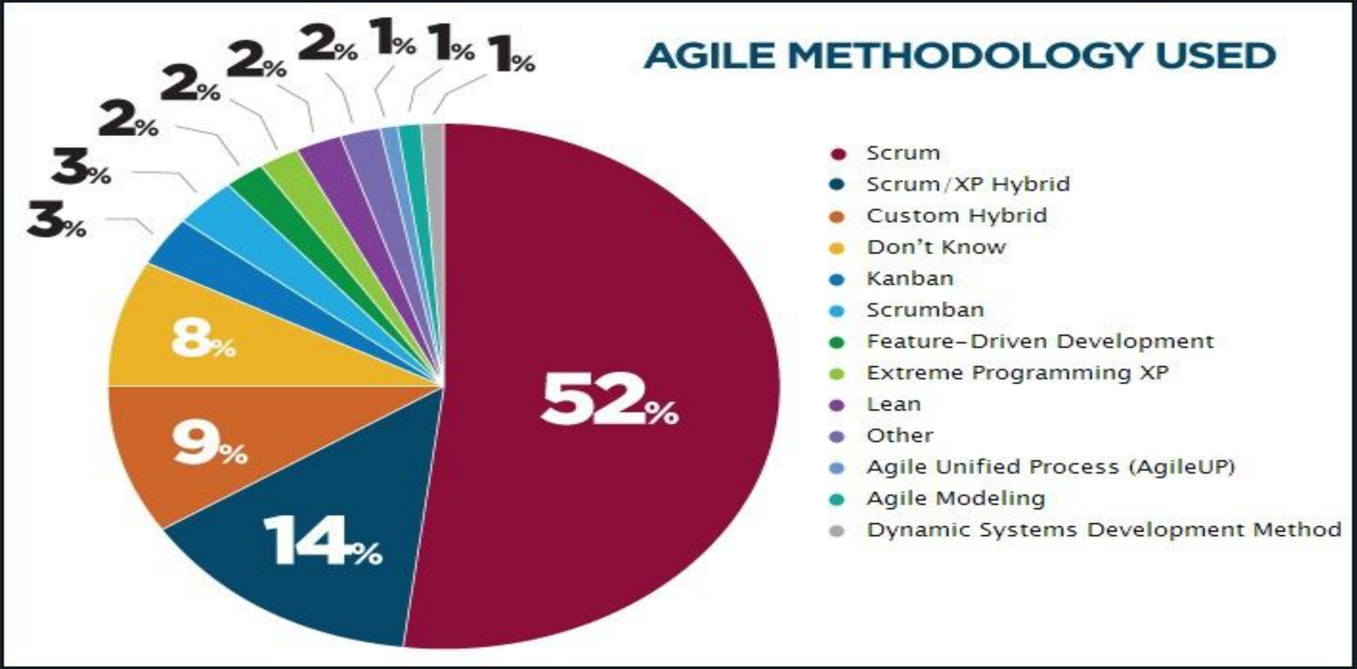


Well Known Agile Methods



- Scrum
- Crystal Methodologies
- DSDM (Dynamic Software Development Method)
- Feature driven development (FDD)
- Lean software development
- Extreme Programming (XP)

Agile Methods Usage Chart



Introduction to Scrum Methodology

- Scrum is an agile process that allows us to focus on delivering the highest business value in the shortest time.

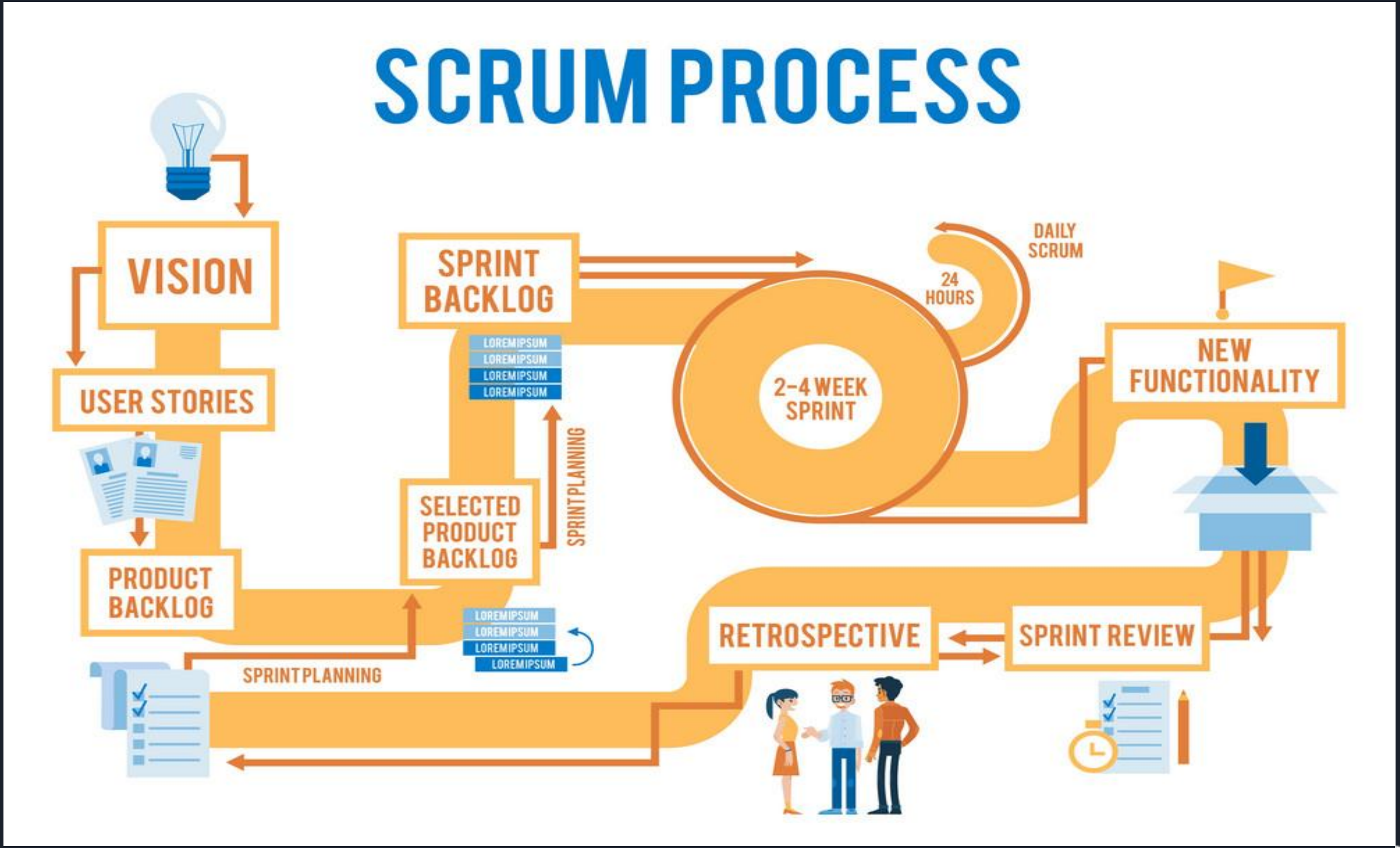
- It allows us to rapidly and repeatedly inspect actual working software (every two weeks to one month).

- The business sets the priorities. Teams self-organize to determine the best way to deliver the highest priority features.

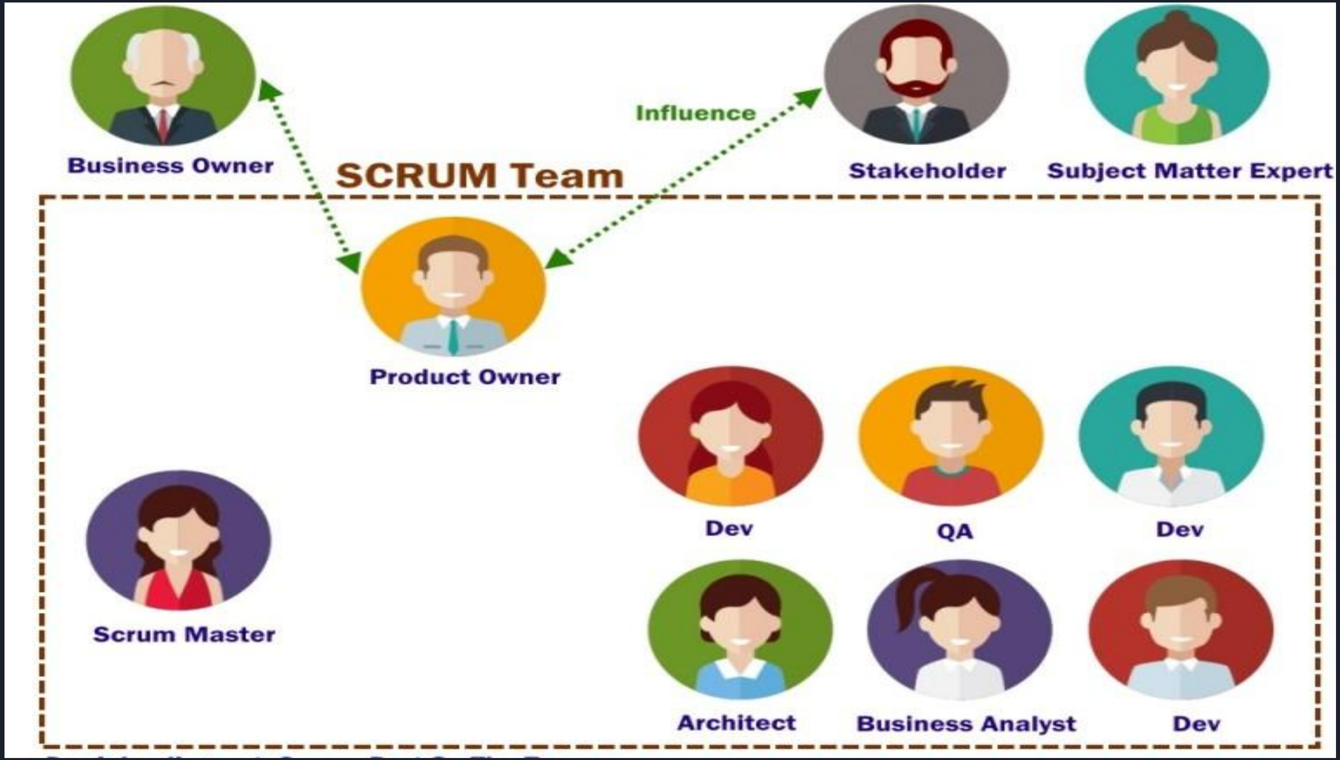
- Every two weeks to a month anyone can see real working software and decide to release it as is or continue to enhance it for another sprint

Scrum Characteristics

- Self-organizing teams
- Product progresses in a series of month-long “sprints”
- Requirements are captured as items in a list of “product backlog”
- No specific engineering practices prescribed
- Uses generative rules to create an agile environment for delivering projects
- One of the “agile processes”

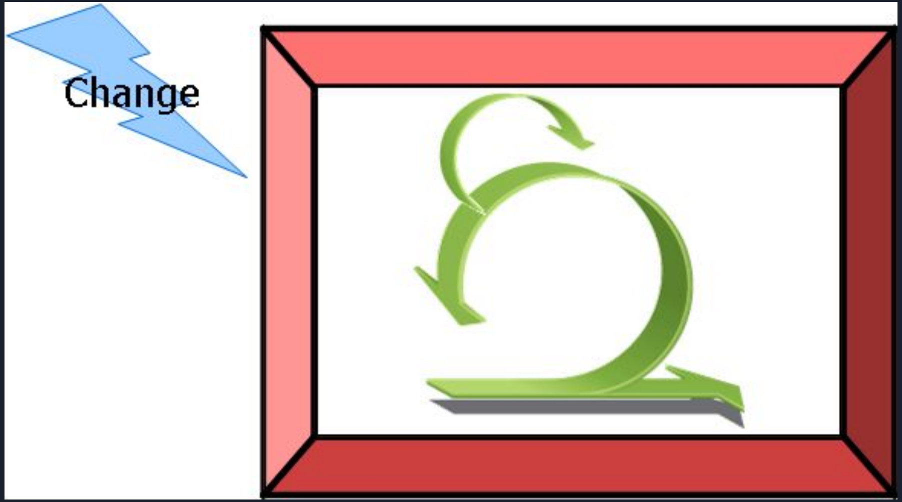

SCRUM PROCESS
VISION
USER STORIES
PRODUCT BACKLOG
SPRINT PLANNING
SELECTED PRODUCT BACKLOG
SPRINT PLANNING
SPRINT BACKLOG
LOREMIPSUM
LOREMIPSUM
LOREMIPSUM
LOREMIPSUM
LOREMIPSUM
LOREMIPSUM
LOREMIPSUM
LOREMIPSUM
2-4 WEEK SPRINT
24 HOURS
DAILY SCRUM
NEW FUNCTIONALITY
SPRINT REVIEW
RETROSPECTIVE

Scrum Team & Roles



No Changes During sPrint

- Plan sprint durations around how long you can commit to keeping change out of the sprint



Roles

- Scrum Master
- Scrum Team
- Product Owner

Events

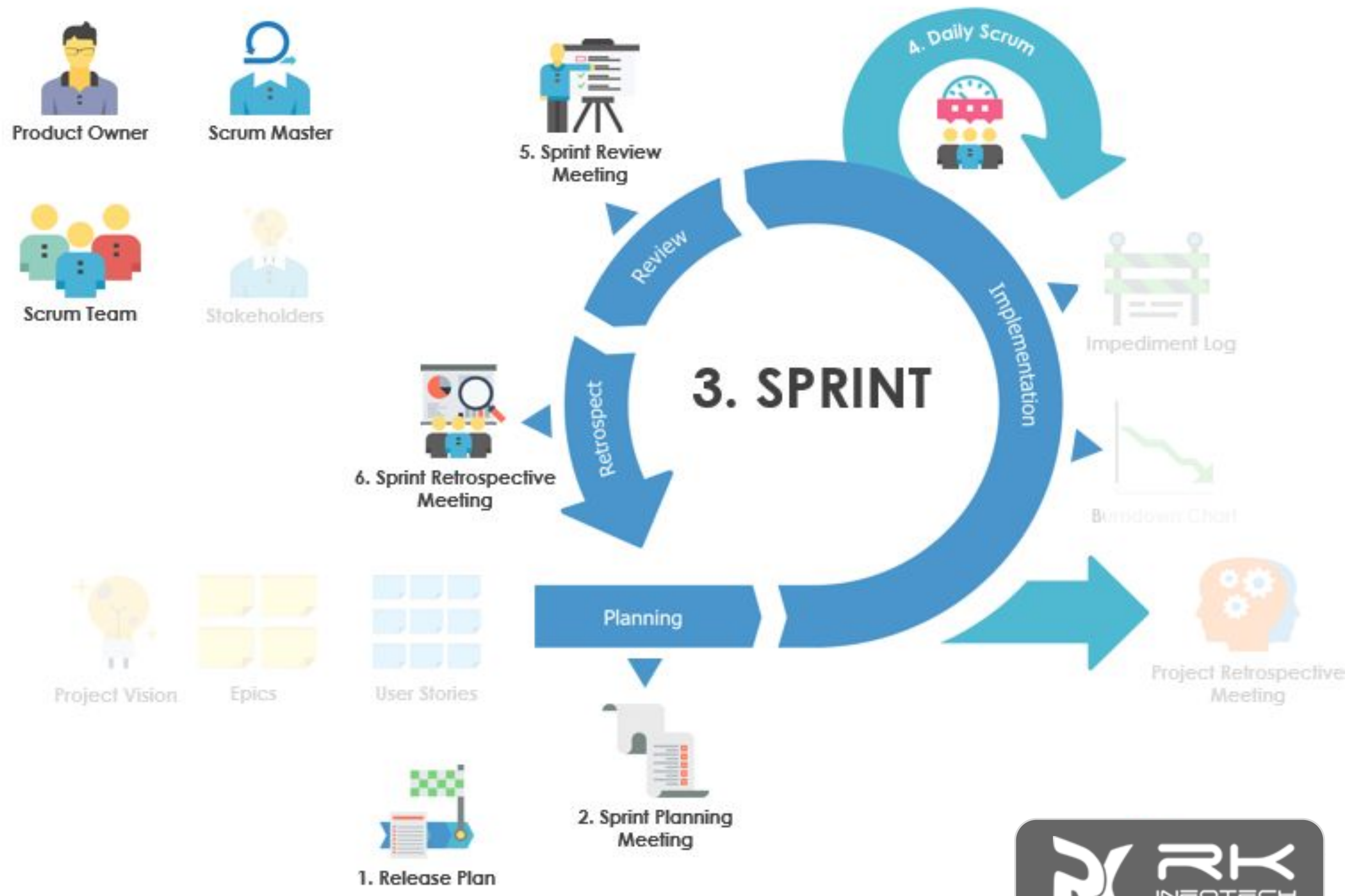
- Release Planning Meeting
- Sprint Planning Meeting
- Sprint
- Daily Scrum
- Sprint Review
- Sprint Retrospective

Artifacts

- Product Backlog
- Sprint Backlog
- Sprint Burndown
- Release Burndown

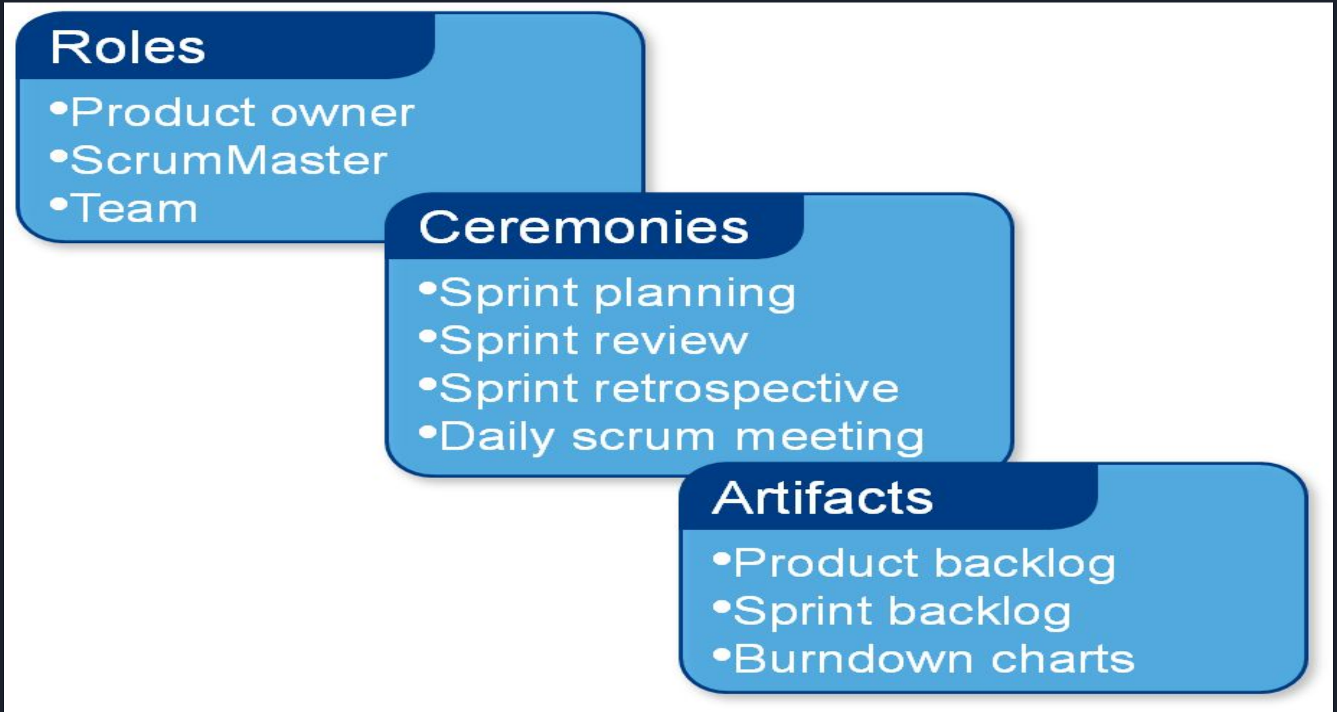
Rules

(bind the model together)



RK INFOTECH
Unwinding IT Solution

Scrum Framework



Roles

- Product owner
- ScrumMaster
- Team

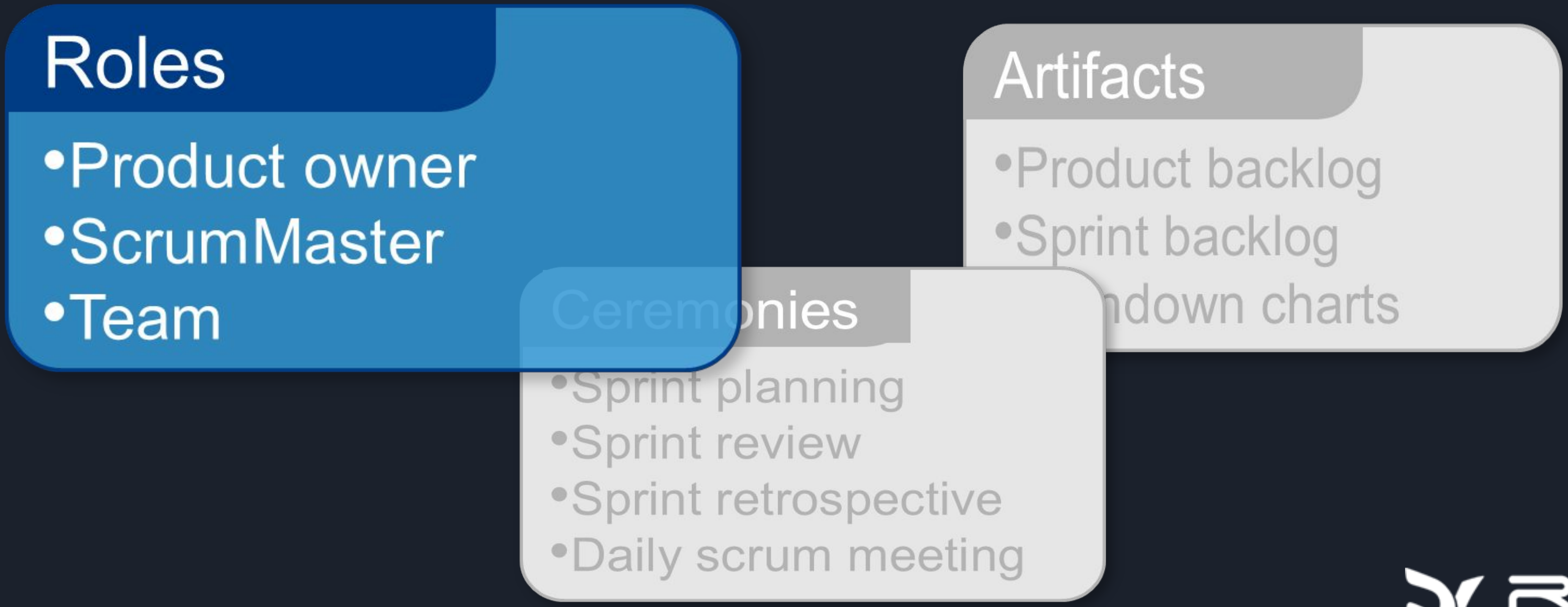
Ceremonies

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

Artifacts

- Product backlog
- Sprint backlog
- Burndown charts

Scrum Framework [Part 1]



Roles

- Product owner
- ScrumMaster
- Team

Artifacts

- Product backlog
- Sprint backlog
- ndown charts

Ceremonies

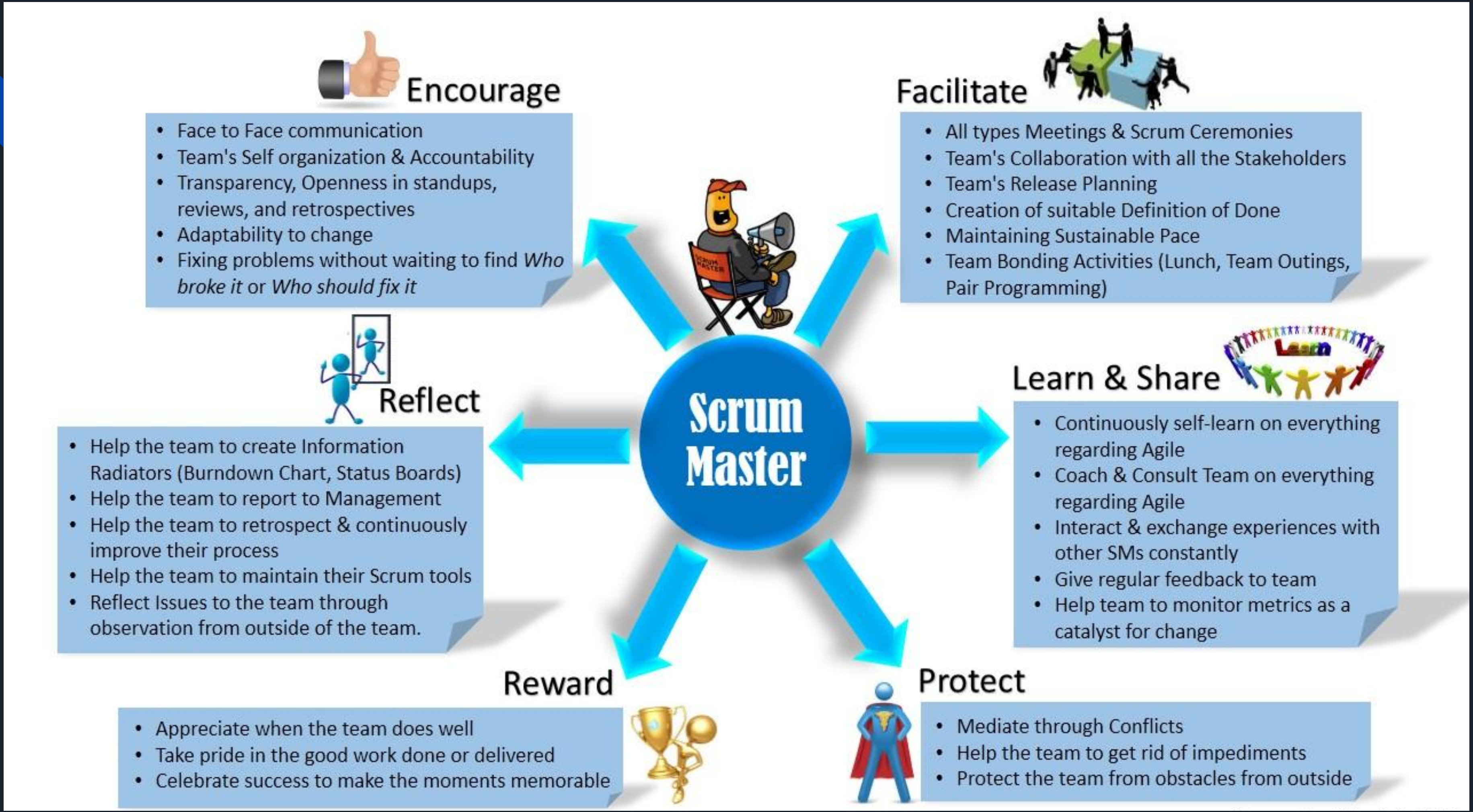
- Sprint planning
- Sprint review
- Sprint retrospective
- Daily scrum meeting

Product Owner

- Define the features of the product
- Decide on release date and content
- Be responsible for the profitability of the product’s return of investment (ROI)
- Prioritize features according to market value
- Adjust features and priority every iteration, as needed
- Accept or reject work results

The ScrumMaster

- Represents management to the project
- Responsible for enacting Scrum values and practices
- Removes obstruction in doing something
- Ensure that the team is fully functional and productive
- Enable close cooperation across all roles and functions
- Shield the team from external interferences



Scrum Master

Encourage

- Face to Face communication
- Team's Self organization & Accountability
- Transparency, Openness in standups, reviews, and retrospectives
- Adaptability to change
- Fixing problems without waiting to find *Who broke it* or *Who should fix it*

Facilitate

- All types Meetings & Scrum Ceremonies
- Team's Collaboration with all the Stakeholders
- Team's Release Planning
- Creation of suitable Definition of Done
- Maintaining Sustainable Pace
- Team Bonding Activities (Lunch, Team Outings, Pair Programming)

Reflect

- Help the team to create Information Radiators (Burndown Chart, Status Boards)
- Help the team to report to Management
- Help the team to retrospect & continuously improve their process
- Help the team to maintain their Scrum tools
- Reflect Issues to the team through observation from outside of the team.

Learn & Share

- Continuously self-learn on everything regarding Agile
- Coach & Consult Team on everything regarding Agile
- Interact & exchange experiences with other SMs constantly
- Give regular feedback to team
- Help team to monitor metrics as a catalyst for change

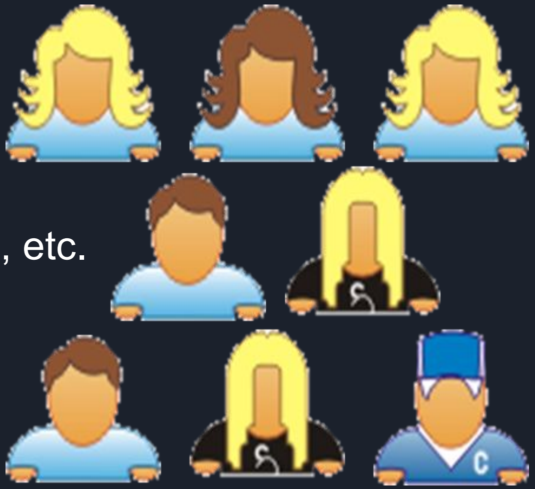
Reward

- Appreciate when the team does well
- Take pride in the good work done or delivered
- Celebrate success to make the moments memorable

Protect

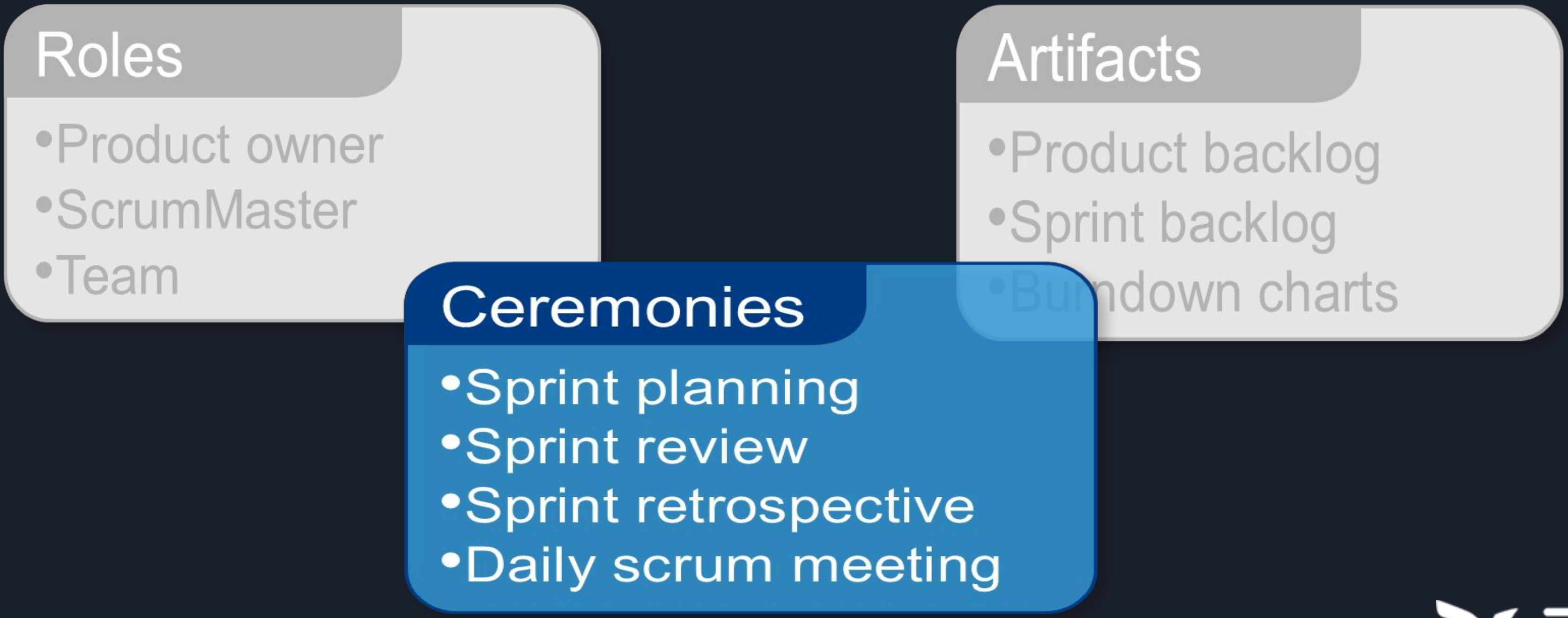
- Mediate through Conflicts
- Help the team to get rid of impediments
- Protect the team from obstacles from outside

The Team



- Typically 5-9 people who are self-organized
- They are Cross-Functional
  - Programmers, testers, user experience designers, etc.
- Skilled in whatever is needed for the project
- Members should be full-time
  - May be exceptions (e.g., database administrator)
- Scrum recognized no sub-teams in the development Team
- Membership should change only between sprints

Scrum Framework [Part 2]



Roles

- Product owner
- ScrumMaster
- Team

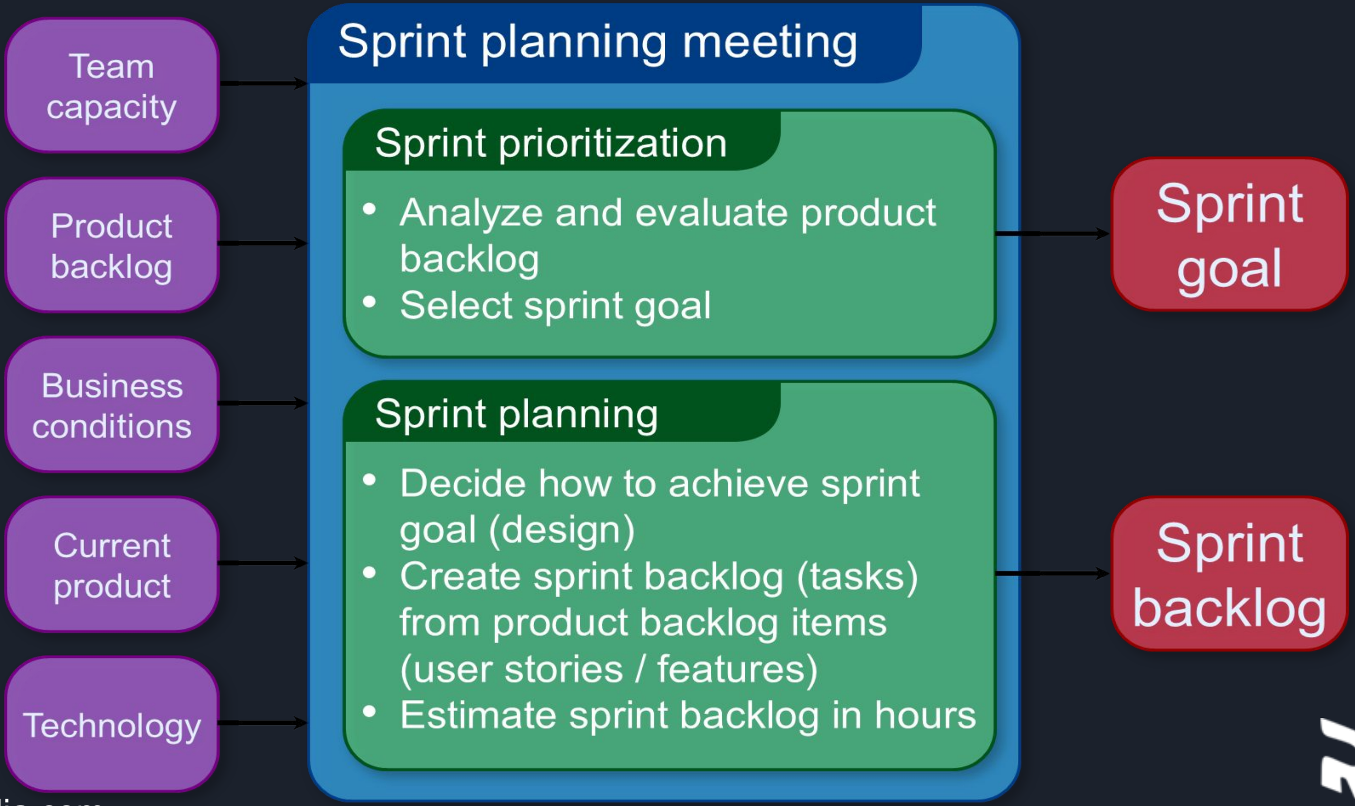
Artifacts

- Product backlog
- Sprint backlog
- Burndown charts

Ceremonies

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

Sprint Planning



Team capacity

Product backlog

Business conditions

Current product

Technology

Sprint planning meeting

Sprint prioritization

- Analyze and evaluate product backlog
- Select sprint goal

Sprint planning

- Decide how to achieve sprint goal (design)
- Create sprint backlog (tasks) from product backlog items (user stories / features)
- Estimate sprint backlog in hours

Sprint goal

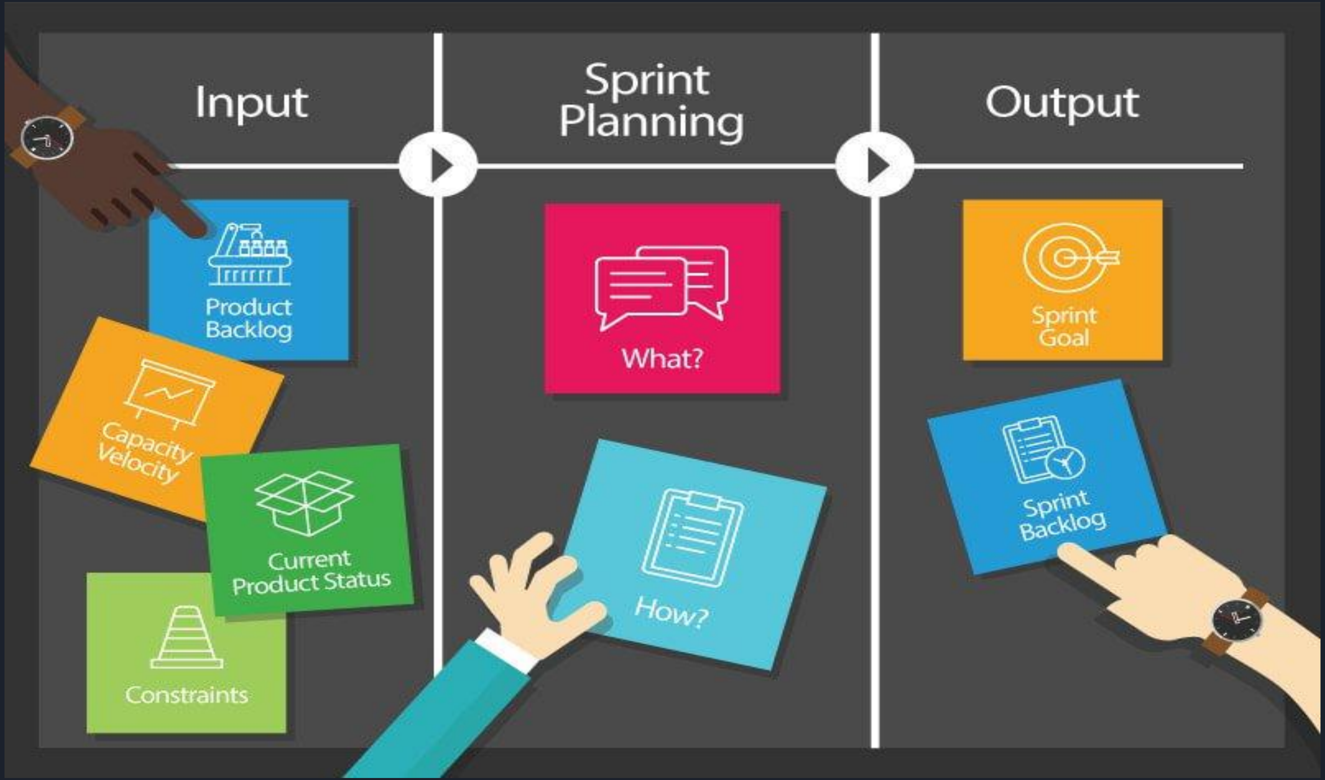
Sprint backlog

Sprint Planning

- Team selects items from the product backlog they can commit to completing
- Sprint backlog is created
- Tasks are identified and each is estimated (1-16 hours)
- Collaboratively, not done alone by the ScrumMaster
- High-level design is considered

- Code the middle tier (8 hours)
- Code the user interface (4)
- Write test fixtures (4)
- Code the foo class (6)
- Update performance tests (4)

Sprint Planning

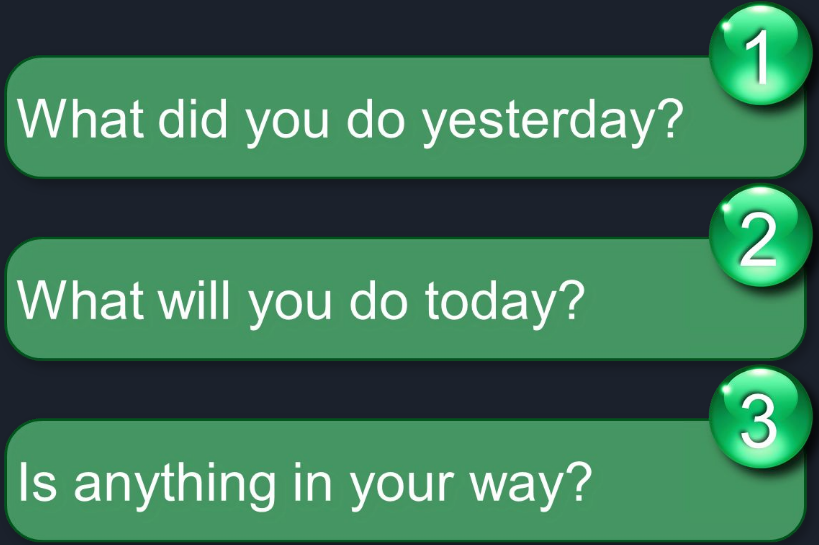


The Daily Scrum

- Daily Activity Tracking
- Quick Problem Solution
- Daily 15-minutes Stand-up
- Not for problem solving
  - Whole world is invited
  - Only team members, ScrumMaster, product owner, can talk
- Project Current Status Analysis
- Helps avoid other unnecessary meetings

Everyone answers 3 questions

- These are not status for the ScrumMaster
  - They are just team member’s commitments in front of other peers



1. What did you do yesterday?
2. What will you do today?
3. Is anything in your way?

The Sprint Review

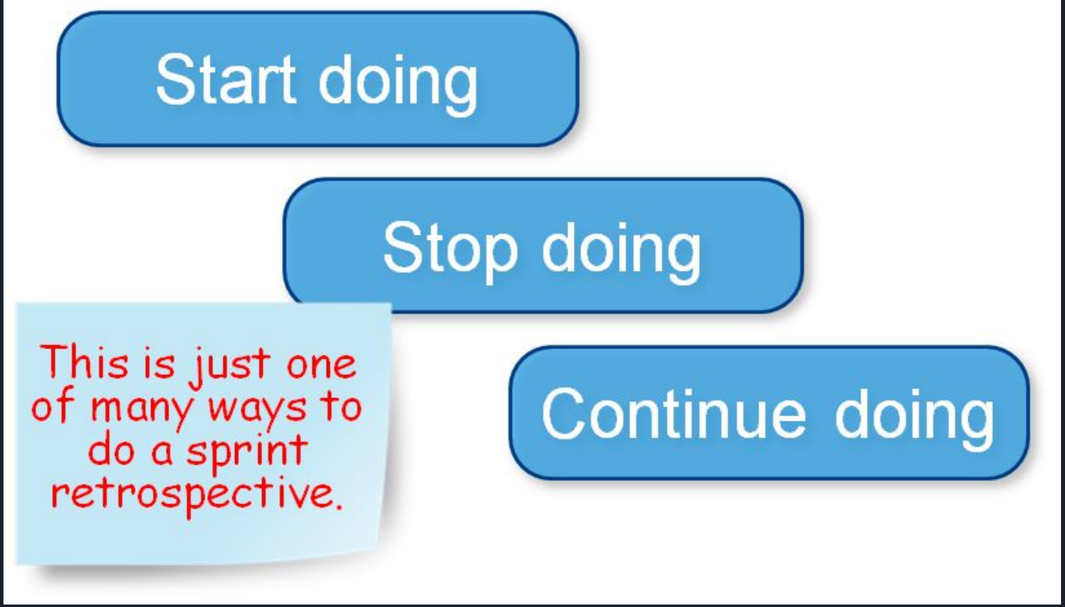
- Team presents what it accomplished during the sprint
- Typically takes the form of a demo of new features or underlying architecture
- Informal Meeting
  - 2-hour prep time rule
  - No slides
- Whole team participates
- Invite the world

Sprint Retrospective

- Periodically take a look at what is and is not working
- Typically 15–30 minutes
- Done after every sprint
- Whole team participates
  - ScrumMaster
  - Product owner
  - Team
  - Possibly customers and others

Sprint Retrospective

- Whole team gathers and discusses what they'd like to :



Start doing

Stop doing

Continue doing

This is just one of many ways to do a sprint retrospective.

Scrum Framework [Part 3]

Roles

- Product owner
- ScrumMaster
- Team

Ceremonies

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

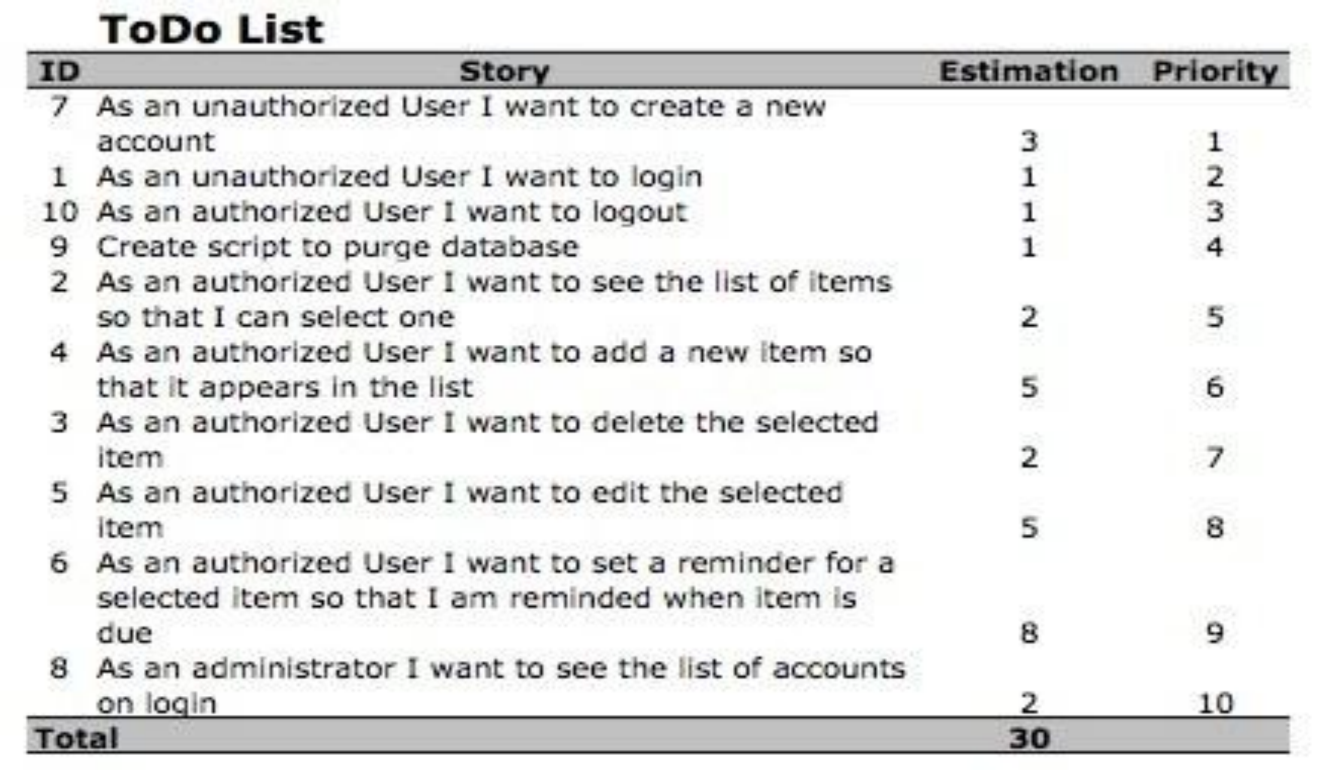
Artifacts

- Product backlog
- Sprint backlog
- Burndown charts

Product Backlog

- The requirements
- A list of all desired work on the project
- Ideally expressed such that each item has value to the users or customers of the product
- Prioritized by the product owner
- Reprioritized at the start of each sprint

Product Backlog Example



ToDo List

ID	Story	Estimation	Priority
7	As an unauthorized User I want to create a new account	3	1
1	As an unauthorized User I want to login	1	2
10	As an authorized User I want to logout	1	3
9	Create script to purge database	1	4
2	As an authorized User I want to see the list of items so that I can select one	2	5
4	As an authorized User I want to add a new item so that it appears in the list	5	6
3	As an authorized User I want to delete the selected item	2	7
5	As an authorized User I want to edit the selected item	5	8
6	As an authorized User I want to set a reminder for a selected item so that I am reminded when item is due	8	9
8	As an administrator I want to see the list of accounts on login	2	10
Total		30	

The Sprint Goal

- A short statement of what the work will be focused on during the sprint

For Example :

Database Application

Make the application run on SQL Server in addition to Oracle.

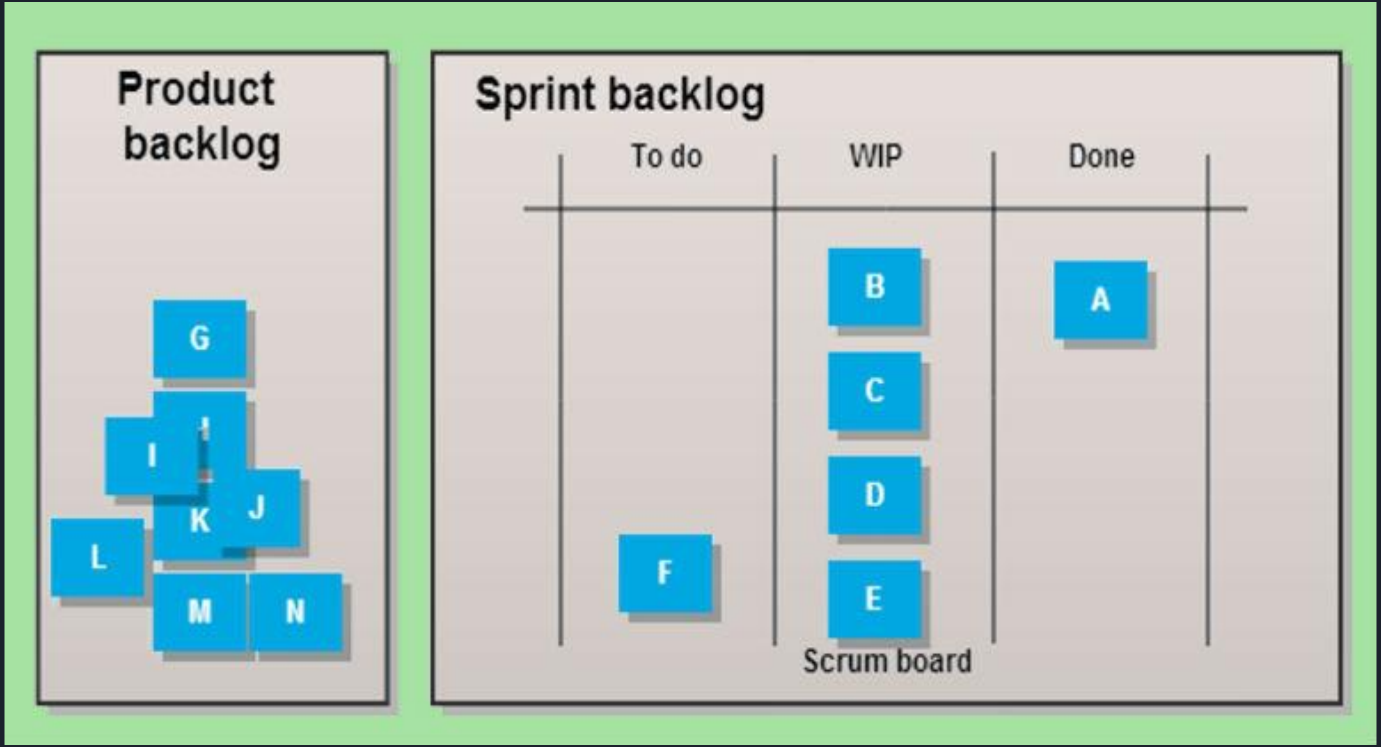
Financial services

Support more technical indicators than company ABC with real-time, streaming data.

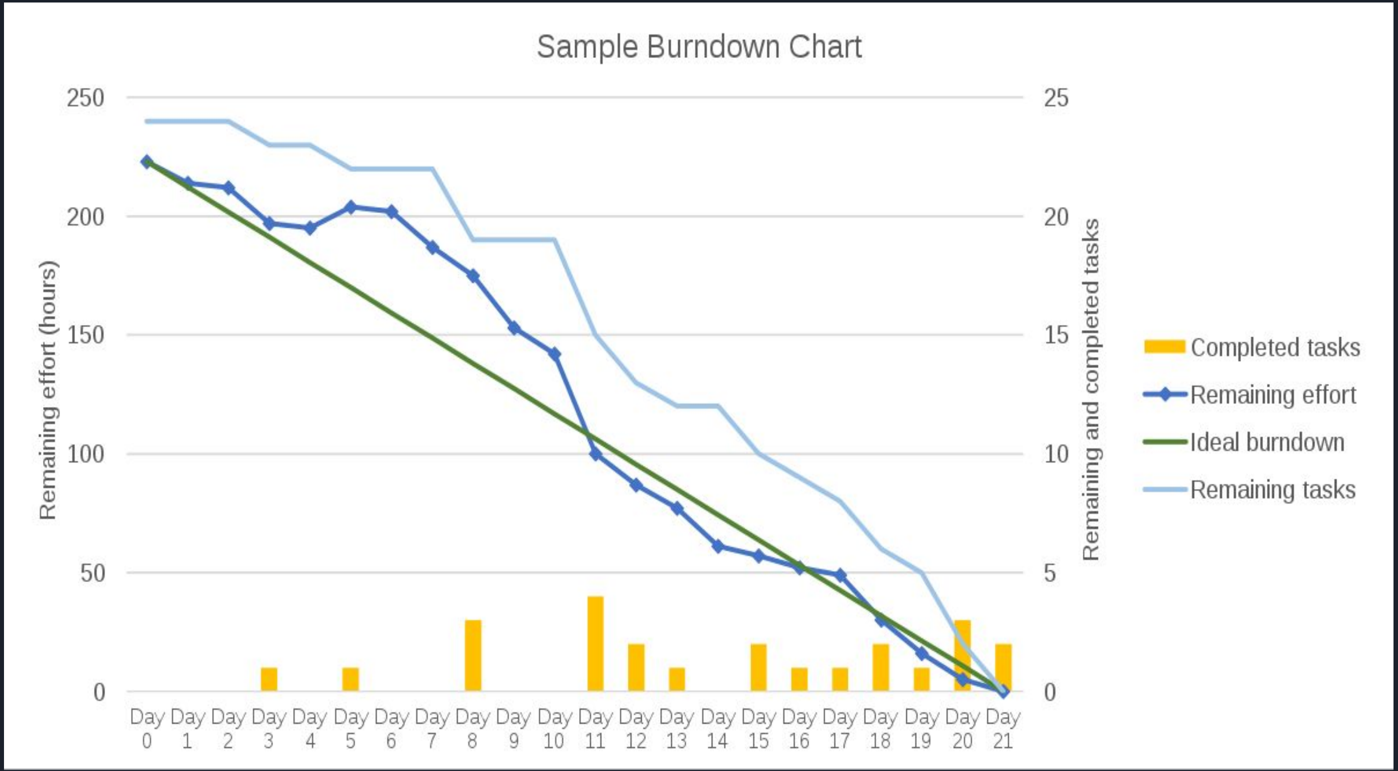
Sprint Backlog

- Individuals sign up for work of their own choosing
  - Work is never assigned
- Estimated work remaining is updated daily
- Any team member can add, delete or change the sprint backlog
- Work for the sprint emerges
- If work is unclear, define a sprint backlog item with a larger amount of time and break it down later
- Update work remaining as more becomes known

Sprint Backlog



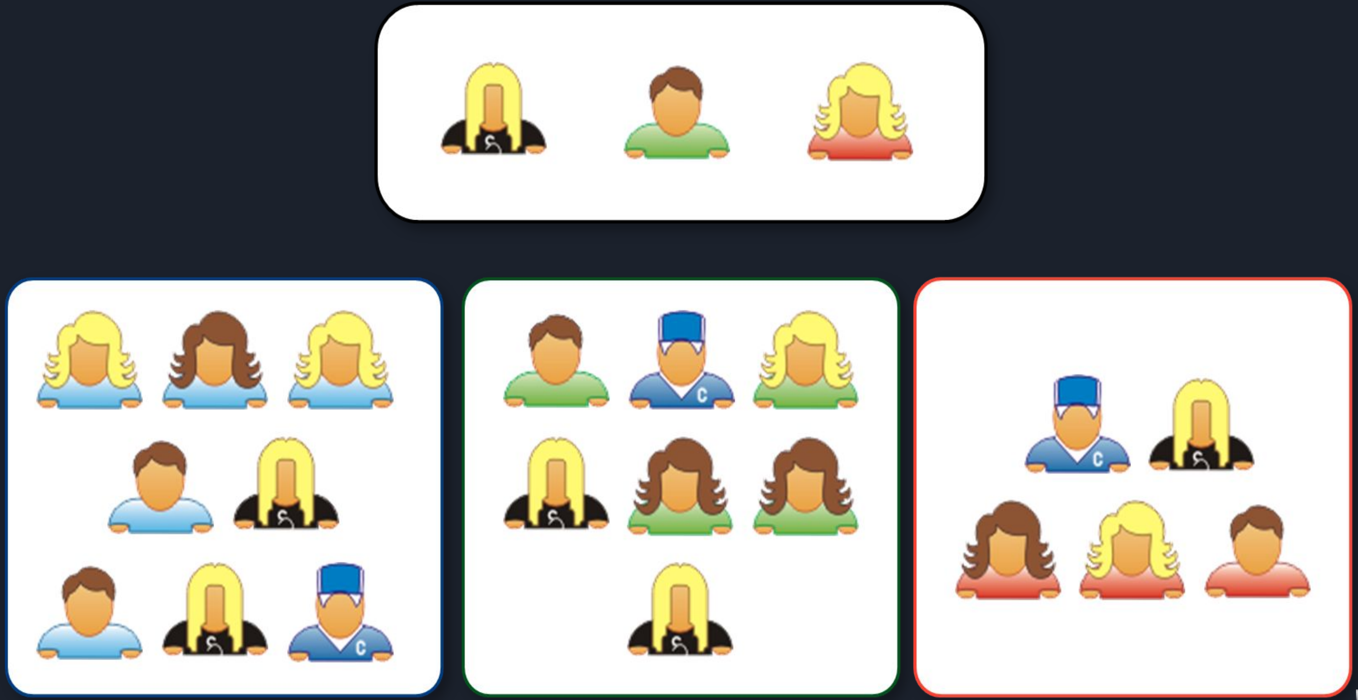
A Sprint Burndown Chart



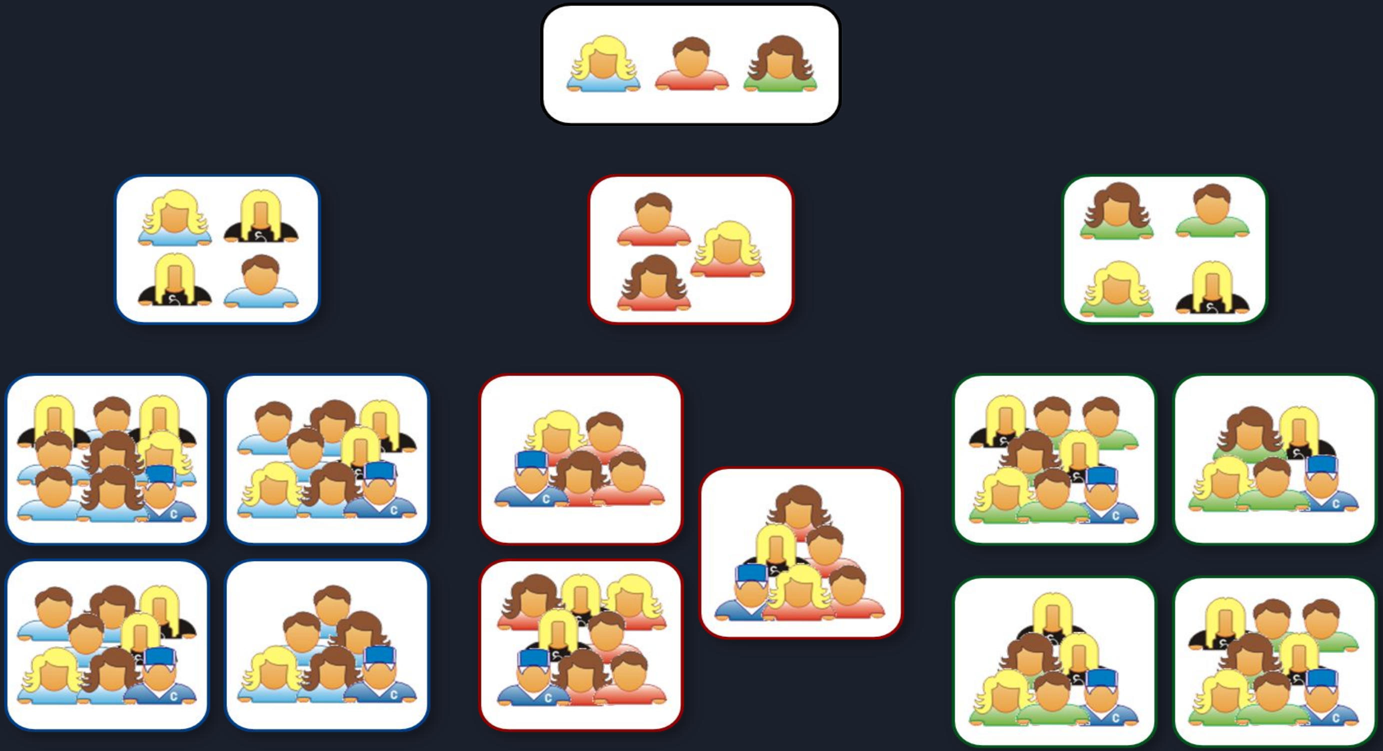
Scalability

- Typical individual team is 7 ± 2 people
  - Scalability comes from teams of teams
- Factors in scaling
  - Type of application
  - Team size
  - Team dispersion
  - Project duration
- Scrum has been used on multiple 500+ person projects

Scaling through the Scrum of scrums



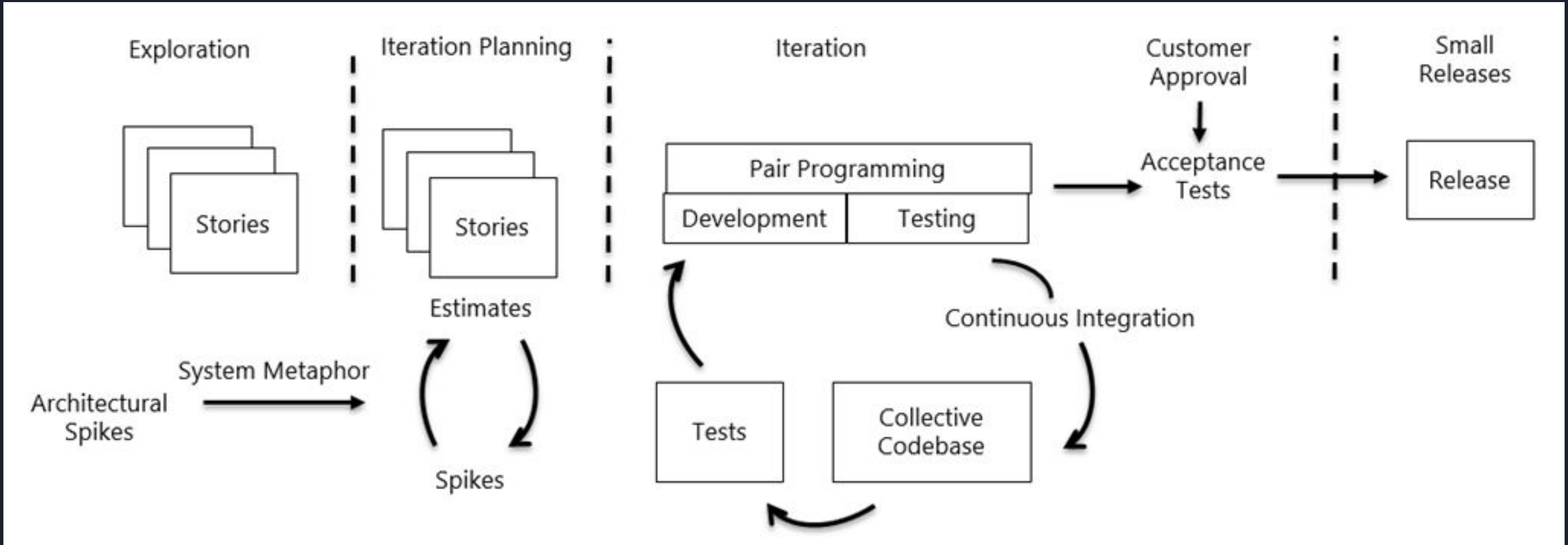
‘N’ Level of Scrums of Scrums



www.rkinfotechindia.com

RK INFOTECH
Unwinding IT Solution

Extreme Programming (XP)

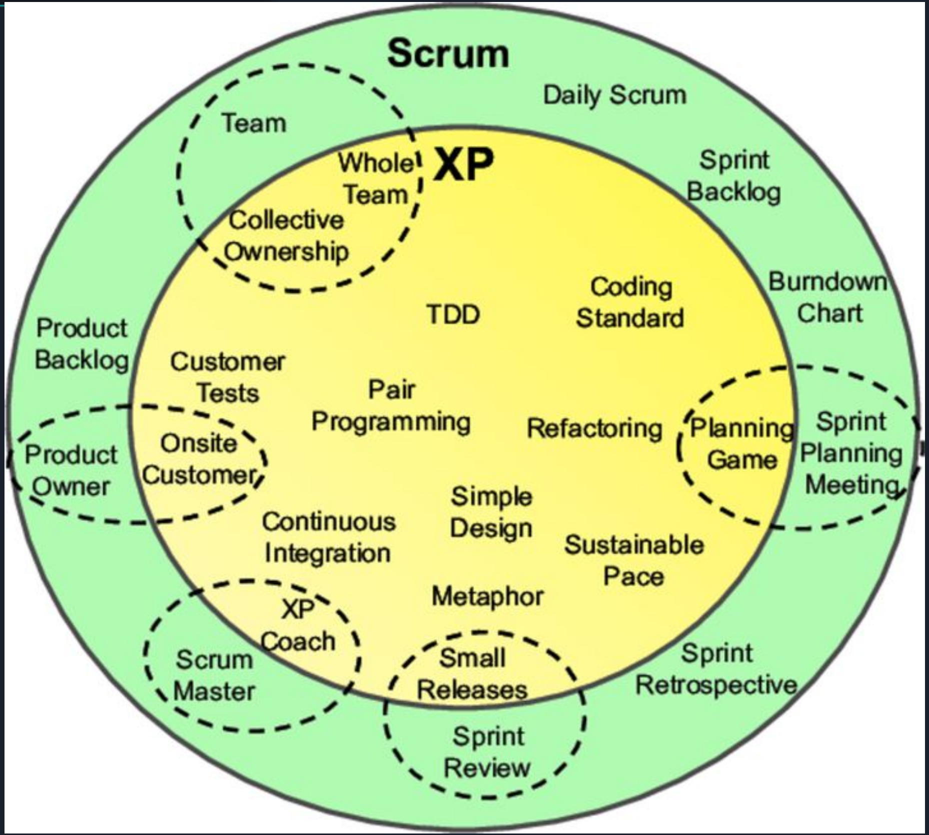


Scrum vs XP

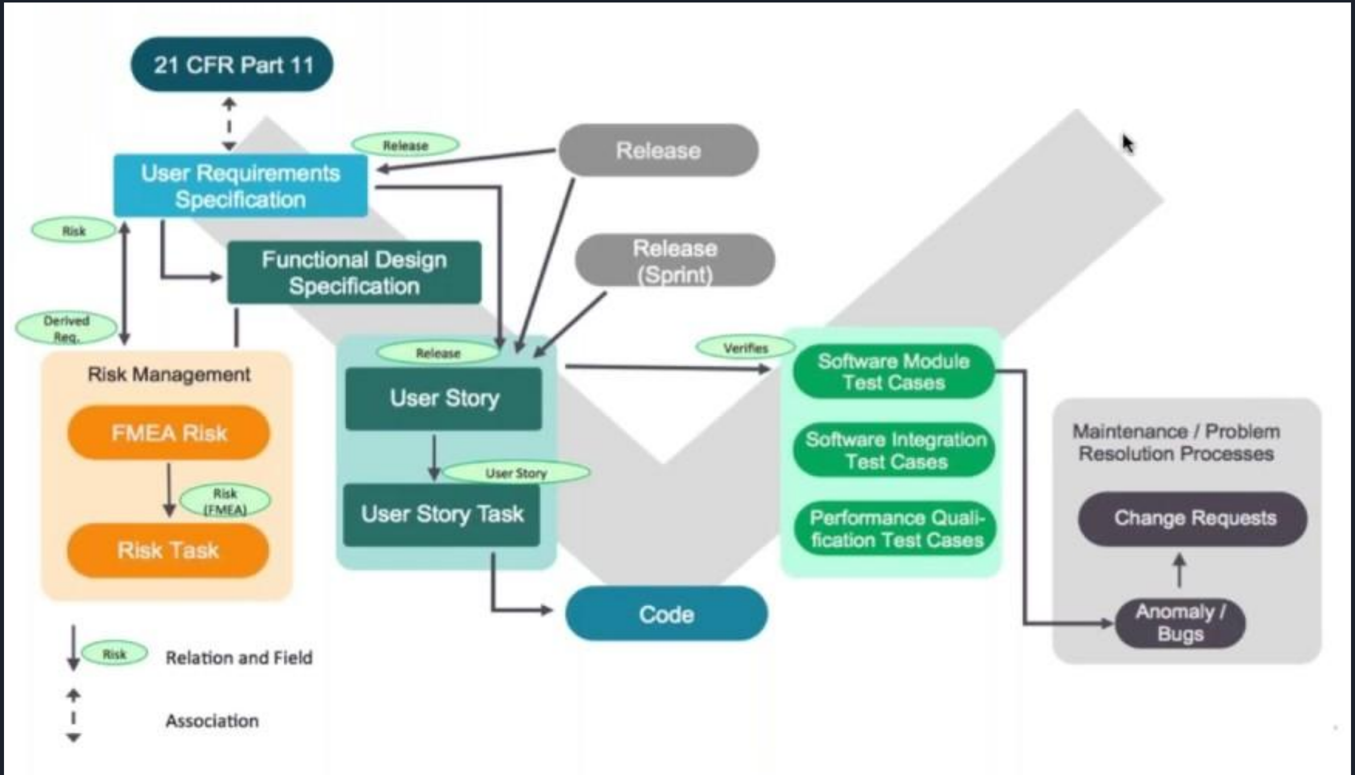
- Scrum teams typically work in iterations (called sprints) that are from two weeks to one month long.
- Scrum product owner prioritizes the product backlog but the team determines the sequence in which they will develop
- Scrum teams do not allow changes into their sprints.
- Scrum doesn't prescribe any engineering practices

- XP teams typically work in iterations that are one or two weeks long.
- XP teams work in a strict priority order.
- XP teams are much more amenable to change within their iterations.
- XP prescribes engineering practices, particularly things like test-driven development, the focus on automated testing, pair programming, simple design, refactoring.

Hybrid Scrum with XP



Hybrid Process Model



RK INFOTECH Unwinding IT Solution

Conclusion with Being Agile

- It is about the people and teams
- It is about customer and delivering software
- It is about continuous improvement

RK INFOTECH
Unwinding IT Solution

Be Agile, not just Do Agile

Thank you!

RK INFOTECH

Unwinding IT Solution

www.rkinfotechindia.com

SF-48 / 49 / 50, 3rd Floor, Shreemad Bhavan,
Opp. Kanta Vikas Gruh, Bhaktinagar Main Road,
Rajkot 360002 (Gujarat) India

(+91) 91735 69325 | (+91) 94271 78991

info@rkinfotechindia.com
careers@rkinfotechindia.com