Introduction to Agile - Scrum

Elements of Agile Adoption

Process

Engineering Practices

Requirement

Don’t build this ...

if all you need is this.

Scrum Overview

- Scrum is a lightweight iterative project framework
- Roles
- Ceremonies
- Artifacts

SCRUM

Scrum

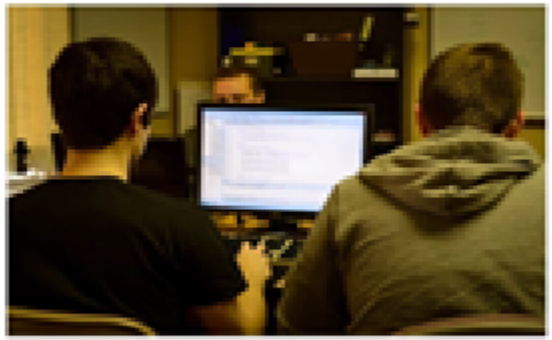
Scrum is an iterative and incremental agile software development framework for managing product development.

It defines "a flexible, holistic product development strategy where a development team works as a unit to reach a common goal".

History of Scrum

- Created by Ken Schwaber and Jeff Sutherland
- Inherited the name Scrum from the paper "The New Product Development Game"

Overview of Scrum



- Commonly used for product development
- Split into 3 areas
  - Roles
  - Ceremonies
  - Artifacts

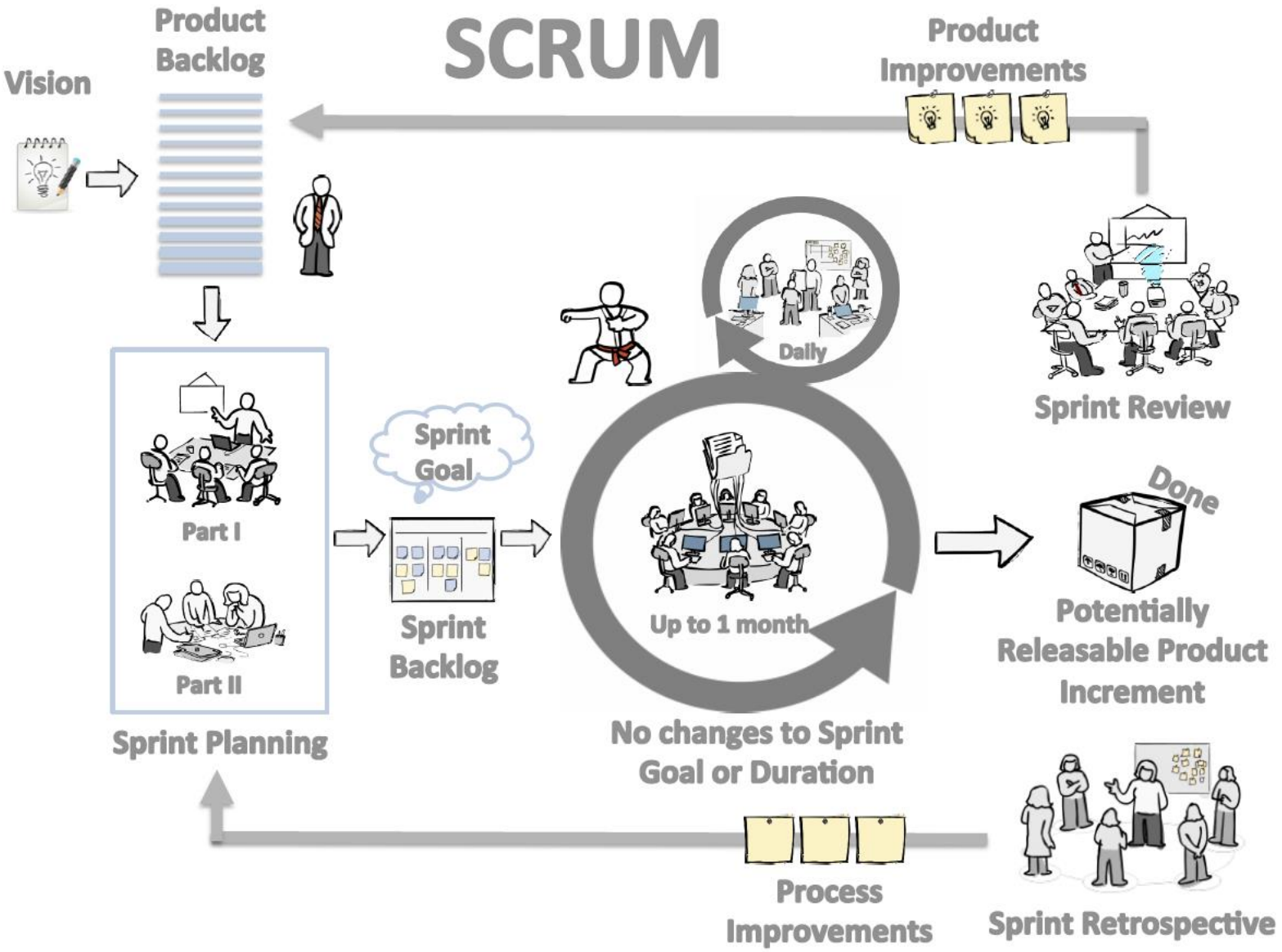
Scrum has been used by:

- Microsoft
- Yahoo
- Google
- Electronic Arts
- High Moon Studios
- Lockheed Martin
- Philips
- Siemens
- Nokia
- Capital One
- BBC
- Intuit
- Intuit
- Nielsen Media
- First American Real Estate
- BMC Software
- Ipswitch
- John Deere
- Lexis Nexis
- Sabre
- Salesforce.com
- Time Warner
- Turner Broadcasting
- Oce

Characteristics

- Scrum Values: **Focus, Respect, Commitment, Openness** and **Courage**.
- Self-organizing teams
- Product progresses in a series of period-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"

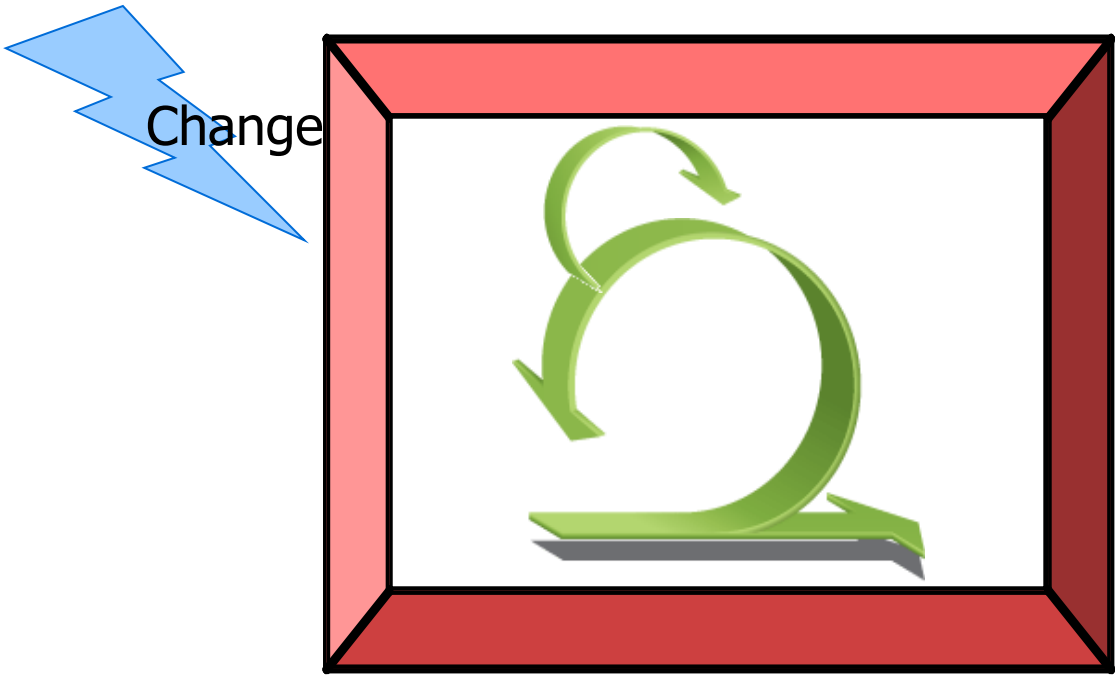
Scrum at a Glance



Sprints

- Scrum projects make progress in a series of "sprints". Fixed Goal, Team, Length
- Typical duration is 2–4 weeks or a calendar month at most
- Why Timeboxing?
  - A constant duration leads to a better rhythm
  - Improve predictability
  - Help stay focused on Goal
- Product is designed, coded, and tested during the sprint

No changes during a sprint



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

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

Roles

- Product owner
- ScrumMaster
- Team

Ceremonies

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

Artifacts

- Product backlog
- Sprint backlog
- Burndown charts

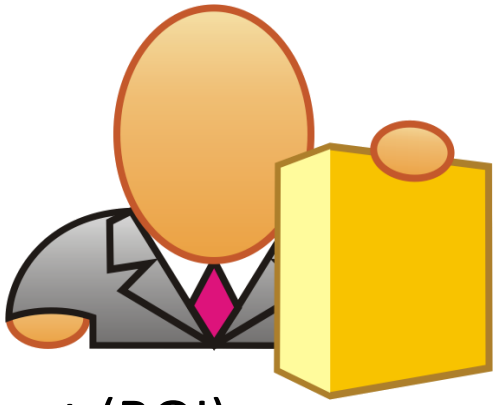
Scrum Roles

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

- Scrum Master
  - Typically a Project & Process Co-ordinator 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

Product owner



- Define the features of the product
- Decide on release date and content
- Be responsible for the profitability of the product (ROI)
- Prioritize features according to market value
- Adjust features and priority every iteration, as needed
- Accept or reject work results
- Responsible for:
  - Product Vision
  - Stakeholder management
  - Scope Management
  - Cost Management
  - Monitoring Release progress

The ScrumMaster

- Responsible for facilitation of all ceremonies
- Responsible for enacting Scrum values and practices
- Removes impediments
- Ensure that the team is fully functional and productive
- Enable close cooperation across all roles and functions
- Shield the team from external interferences

The team

- Typically 5-9 people
- Cross-functional:
  - Programmers, testers, user experience designers, etc.
- Members should be full-time
  - May be exceptions (e.g., database administrator)

The team

- Teams are self-organizing
  - Ideally, no titles but rarely a possibility
- Membership should change only between sprints

Agreement - Definition of “Ready” (DoR)

- For a Story to be “Ready”, following criteria have to be met
  - The story should reasonably show INVEST characteristics

    I – Independent / Immediately actionable

    N – Negotiable

    V – Valuable to the customer, user or product

    E – Estimable

    S – Sized to fit

    T – Testable
  - The business implications of the story have been discussed, any impacts to finance, customer care have been addressed
  - The User Interaction Design is ready (At the very least wireframes covering all interactions of the story should be available)
  - Any design assets needed for the story have been prepared to a reasonable degree (PSDs for some if not all pages in the Story should be available)

Agreement - Definition of “Done” (DoD)

- Definition of Done must describe exactly what “done” means
  - Product Owner must pay careful attention when defining the DoD
  - The scrum team must challenge the DoD, if necessary
  - *“What’s not in DoD, is not needed”*
  - Item is either “*done*” or “*not done*”
- Example:
  - Story: Picture upload
    - end user can upload his/her picture from profile settings page
    - picture is shown on the left upper corner of the profile page
    - picture is scaled to fit the profile picture box on the profile page
    - functional tests are passed
    - regression tests are passed
    - design documents are updated
    - user's guide is updated
- **Does not** define any details of the implementation!

Scrum framework

Roles

- Product owner
- ScrumMaster
- Team

Ceremonies

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

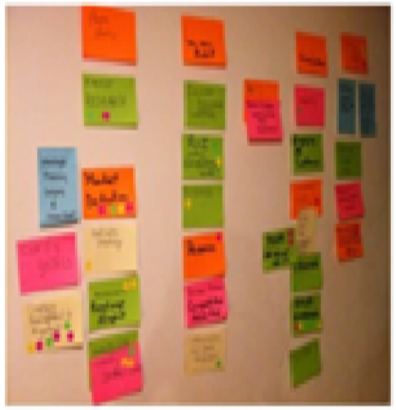
Artifacts

- Product backlog
- Sprint backlog
- Burndown charts

Scrum Ceremonies

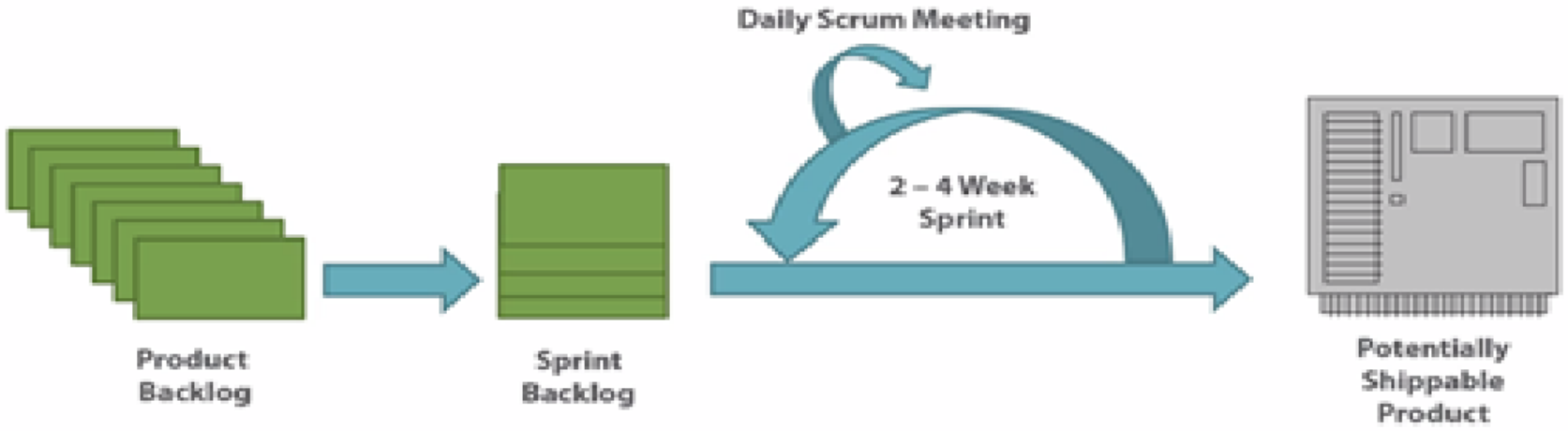
- Sprint planning
- Sprint review
- Sprint retrospective
- Daily scrum meeting

Scrum Artifacts

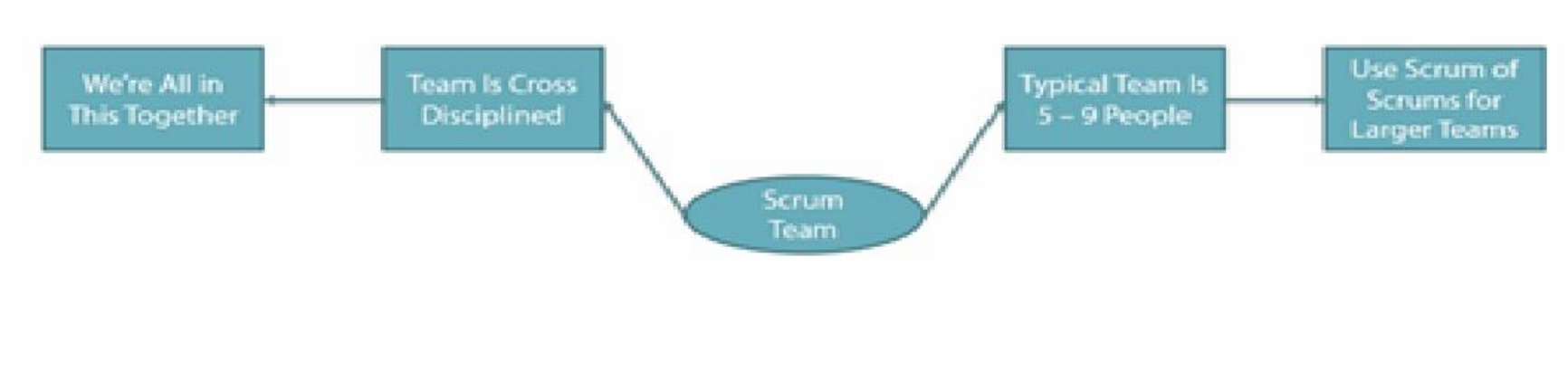


- Product backlog
- Sprint backlog
- Burndown chart

SCRUM DIAGRAM



Scrum Team


We're All in This Together
Team Is Cross Disciplined
Scrum Team
Typical Team Is 5 – 9 People
Use Scrum of Scrums for Larger Teams

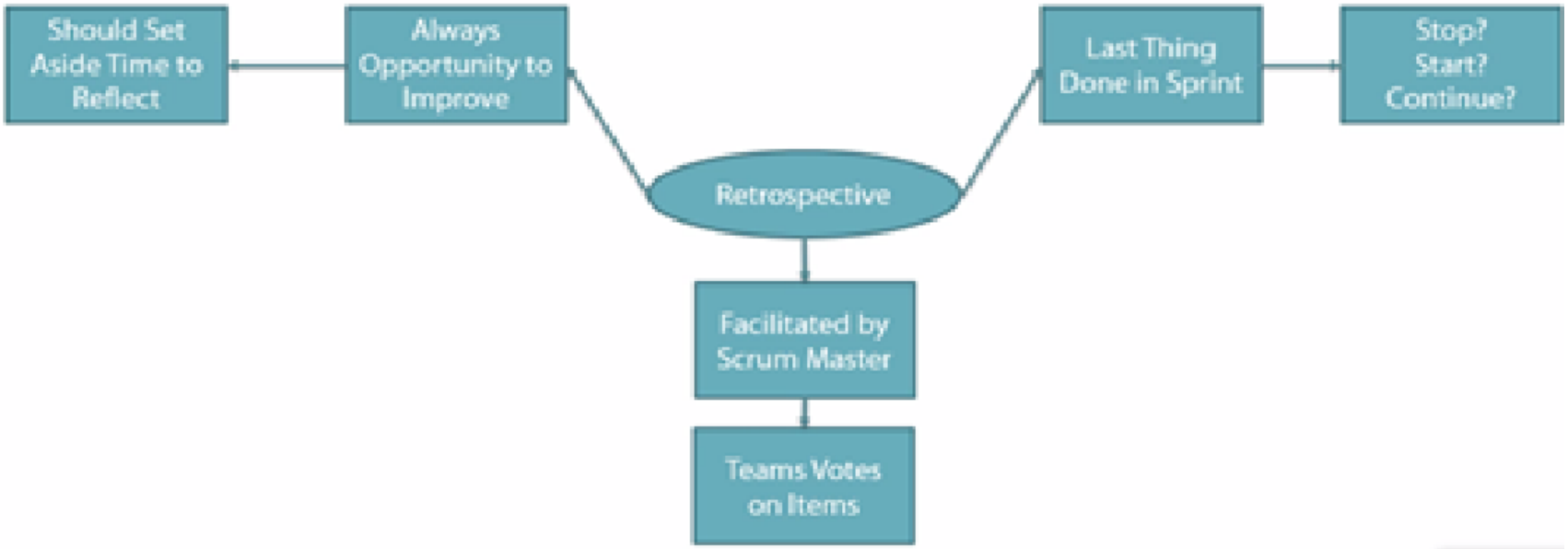
Scrum ceremonies

Planning Meeting, Sprint Review, Sprint Retrospective and the Daily Scrum

Planning Meeting

Sprint Review

Sprint Retrospective



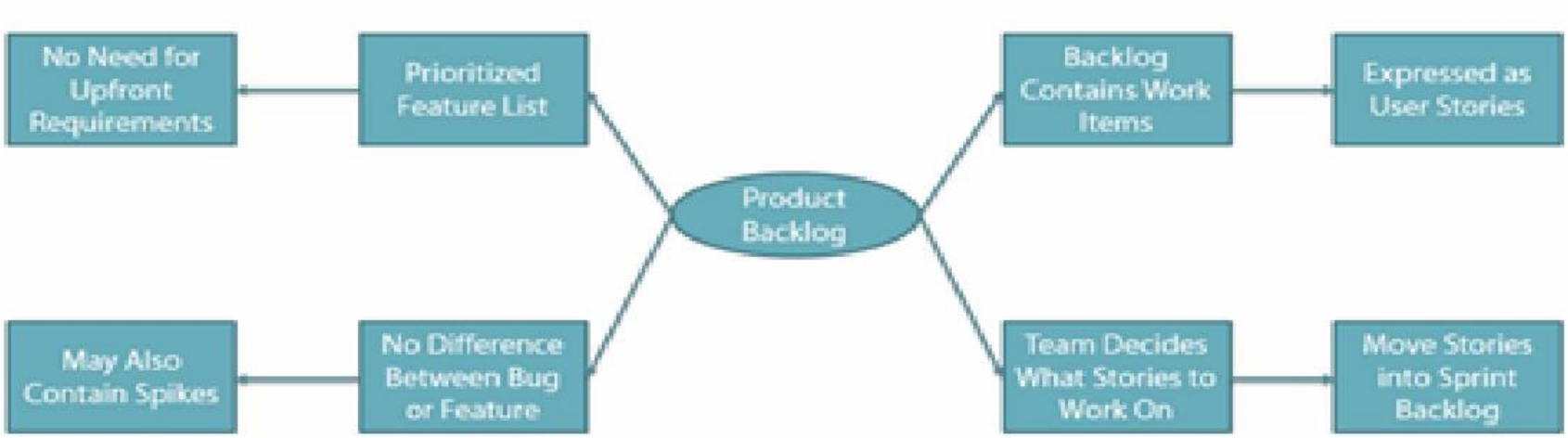
Daily Scrum


Stand in a Circle
Held Everyday
Daily Scrum
Answer 3 Questions
What Did You Achieve Yesterday?
What Will You Achieve Today?
Is Anything Blocking You?

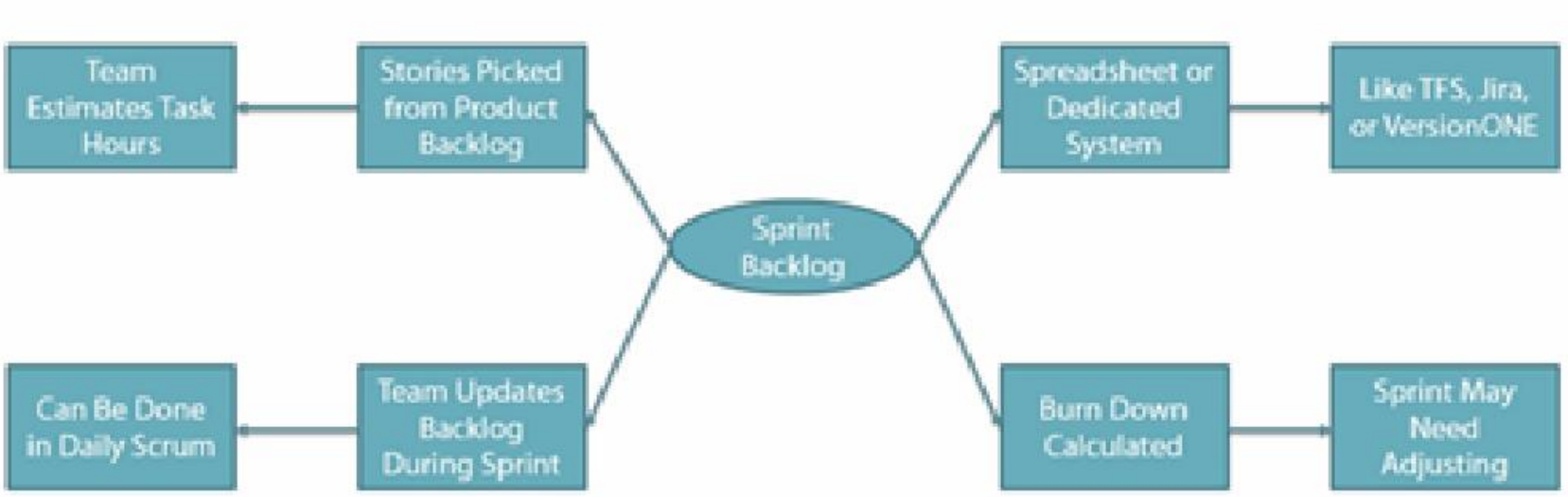
Scrum Artifacts

Product Backlog, Sprint Backlog and the Burn Down Chart

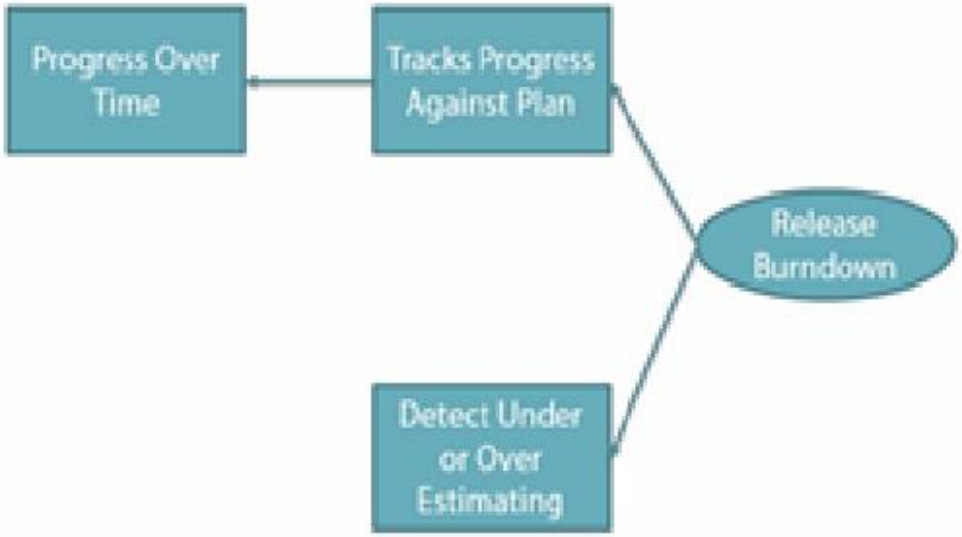
Product Backlog

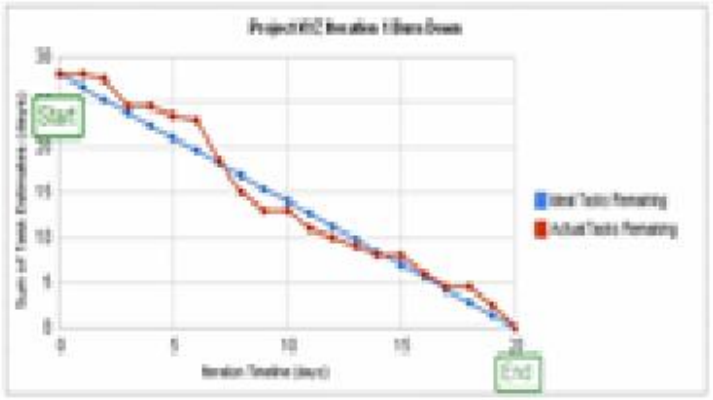


Sprint Backlog

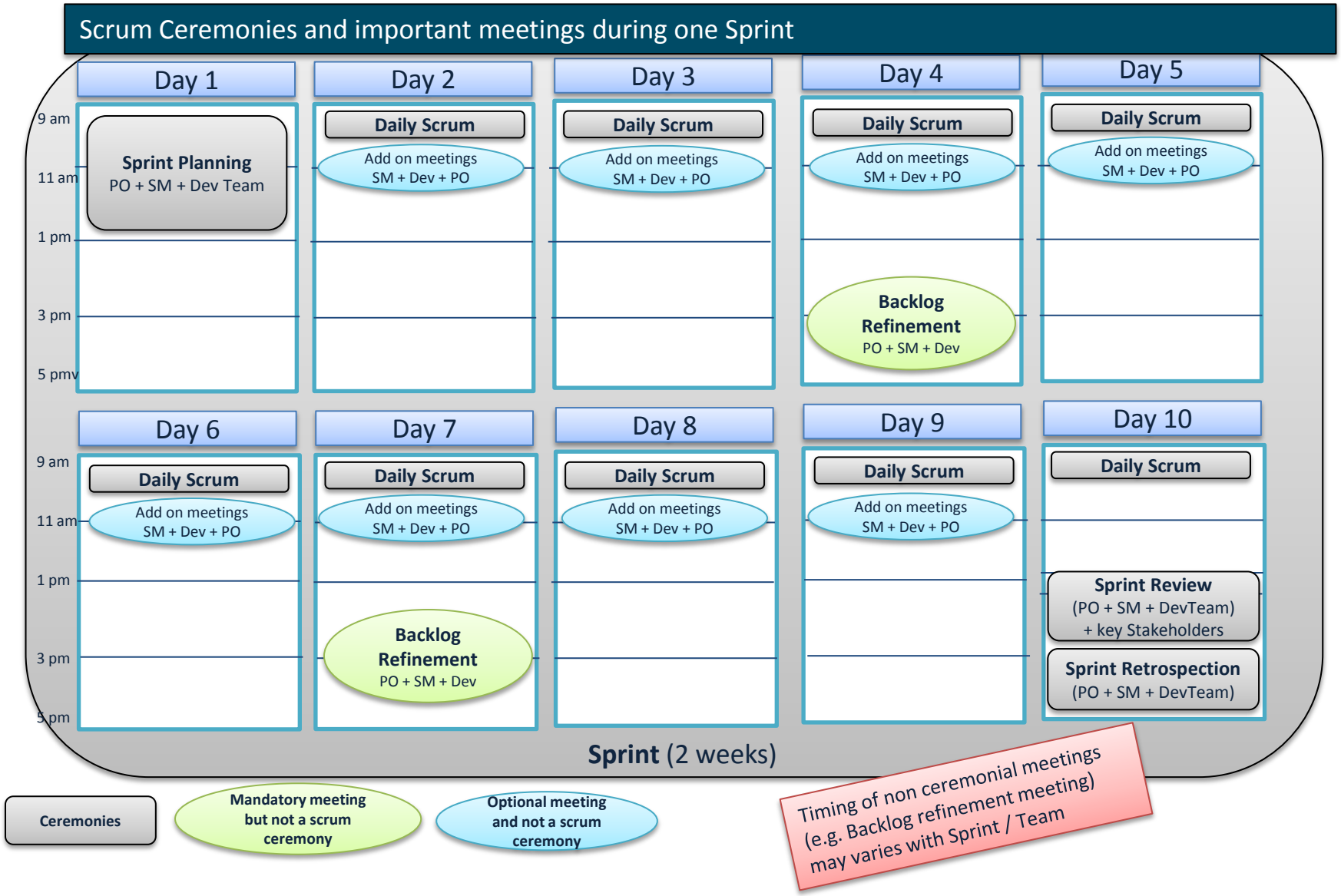


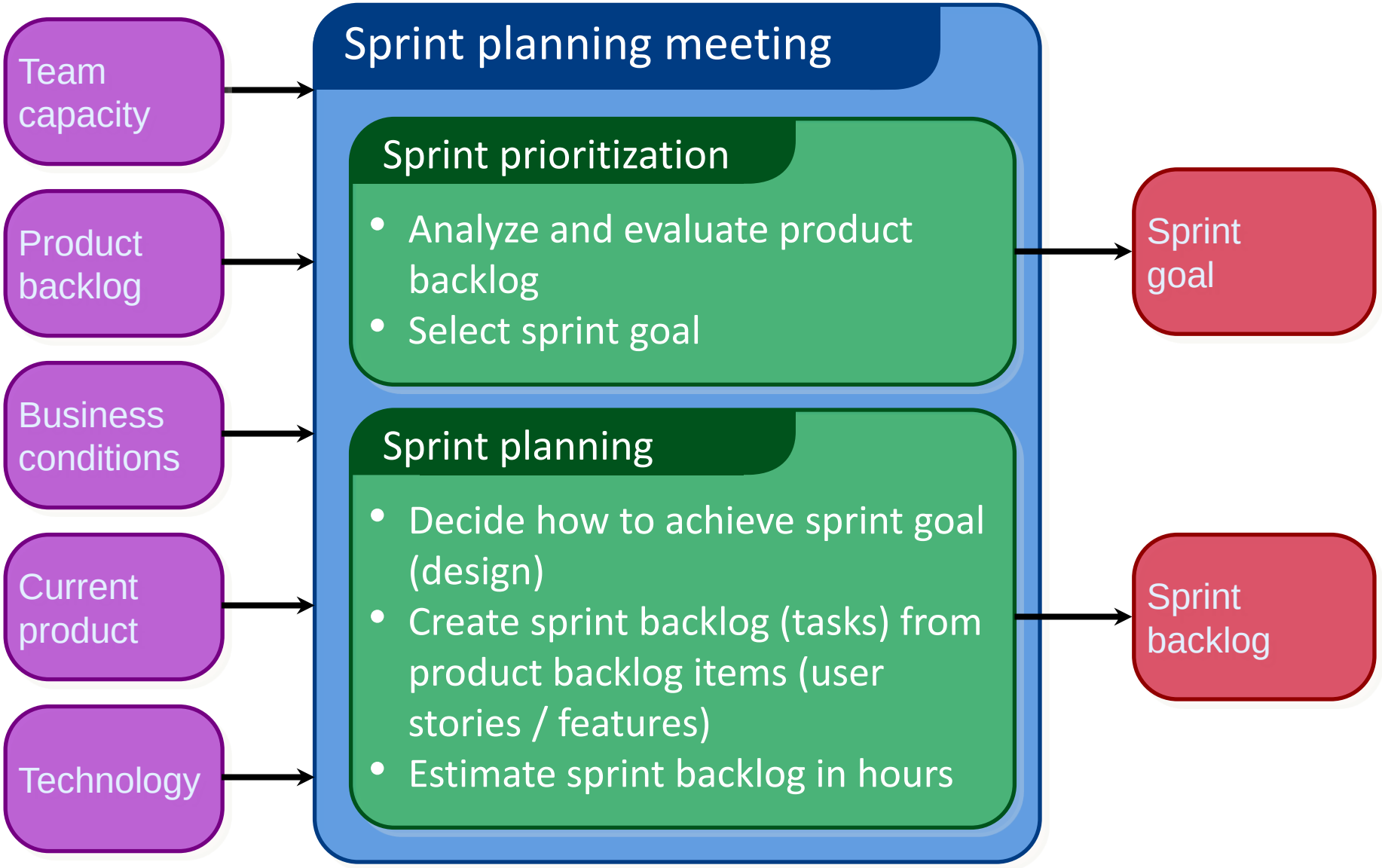
Burn Down Chart





Scrum with 2 weeks Sprint Duration




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

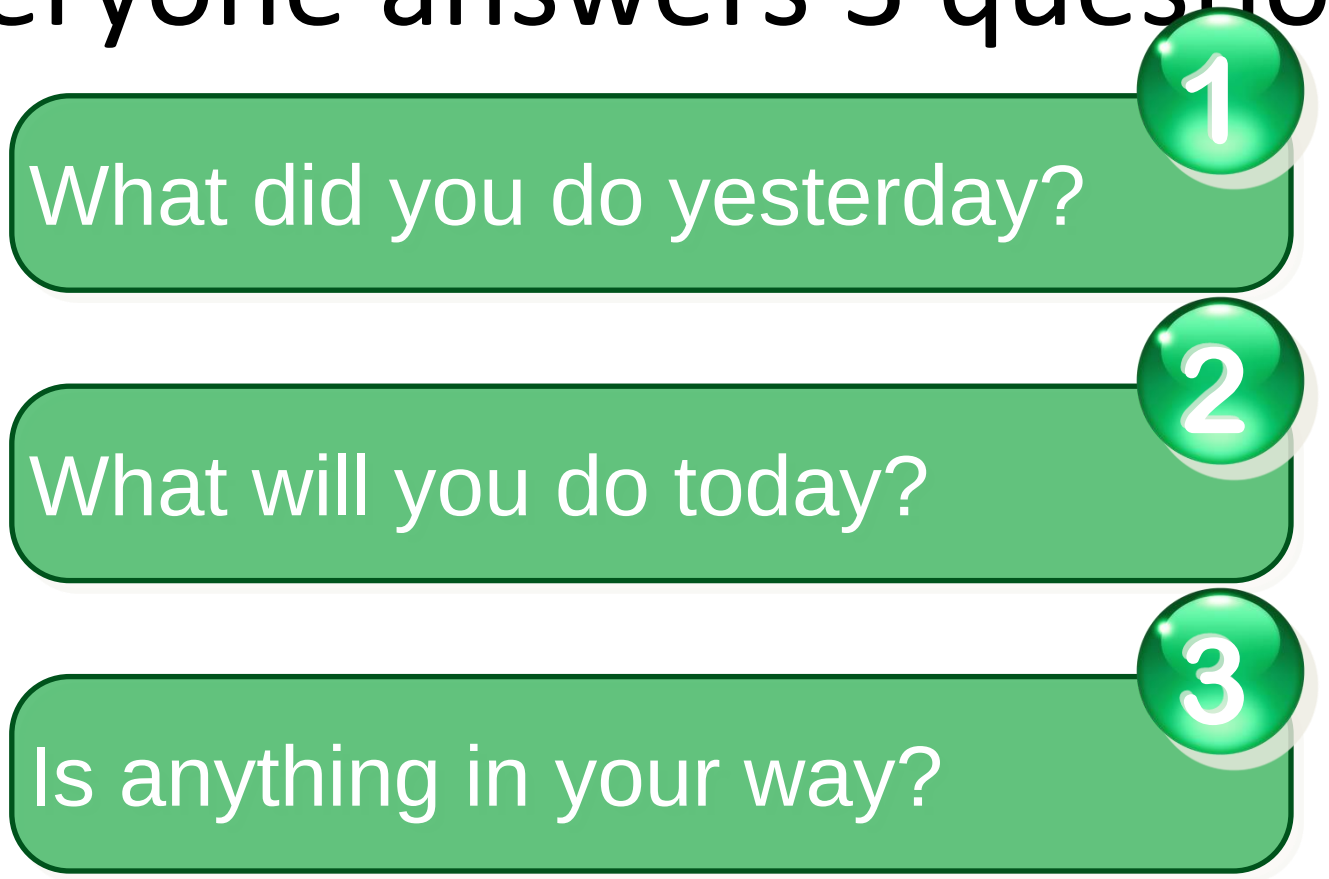
As a vacation planner, I want to see photos of the hotels.

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

The daily scrum

- Parameters
  - Daily
  - 15-minutes
  - Stand-up
- Not for problem solving
  - Whole world is invited
  - Only team members, ScrumMaster, product owner, can talk
- Helps avoid other unnecessary meetings

Everyone answers 3 questions



- These are *not* status for the ScrumMaster
  - They are commitments in front of peers

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
  - 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

Start / Stop / Continue

- 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

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

Sample Product Backlog

Backlog item	Estimate
Allow a guest to make a reservation	3 (story points)
As a guest, I want to cancel a reservation.	5
As a guest, I want to change the dates of a reservation.	3
As a hotel employee, I can run Revenue reports	8
Improve exception handling	8
...	30
...	50

The sprint goal

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

Life Sciences

Support features necessary for population genetics studies.

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 change 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

Sample Sprint Backlog

Sprint 1

01/11/2015

19 days work in this sprint

	Backlog Item	Backlog Item	Owner	Estimate	Sprint Day	1 Mo	2 Tu	3 We	4 Th	5 Fr	6 Sa	7 Su
					Hours remaining	152	152	152	152	152	152	152
1	Minor	Remove user kludge in .dpr file	BC	8		8	8	8	8	8	8	8
2	Minor	Remove cMap/cMenu/cMenuSize from disciplines.pas	BC	8		8	8	8	8	8	8	8
3	Minor	Create "Legacy" discipline node with old civils and E&I content	BC	8		8	8	8	8	8	8	8
4	Major	Augment each tbl operation to support network operation	BC	80		80	80	80	80	80	80	80
5	Major	Extend Engineering Design estimate items to include summaries	BC	16		16	16	16	16	16	16	16
6	Super	Supervision/Guidance	CAM	32		32	32	32	32	32	32	32

Sprint 1

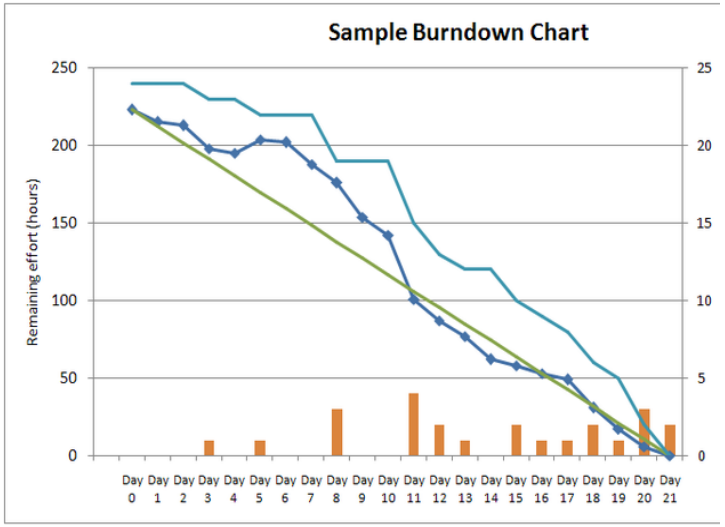
01/11/2015

19 days work in this sprint

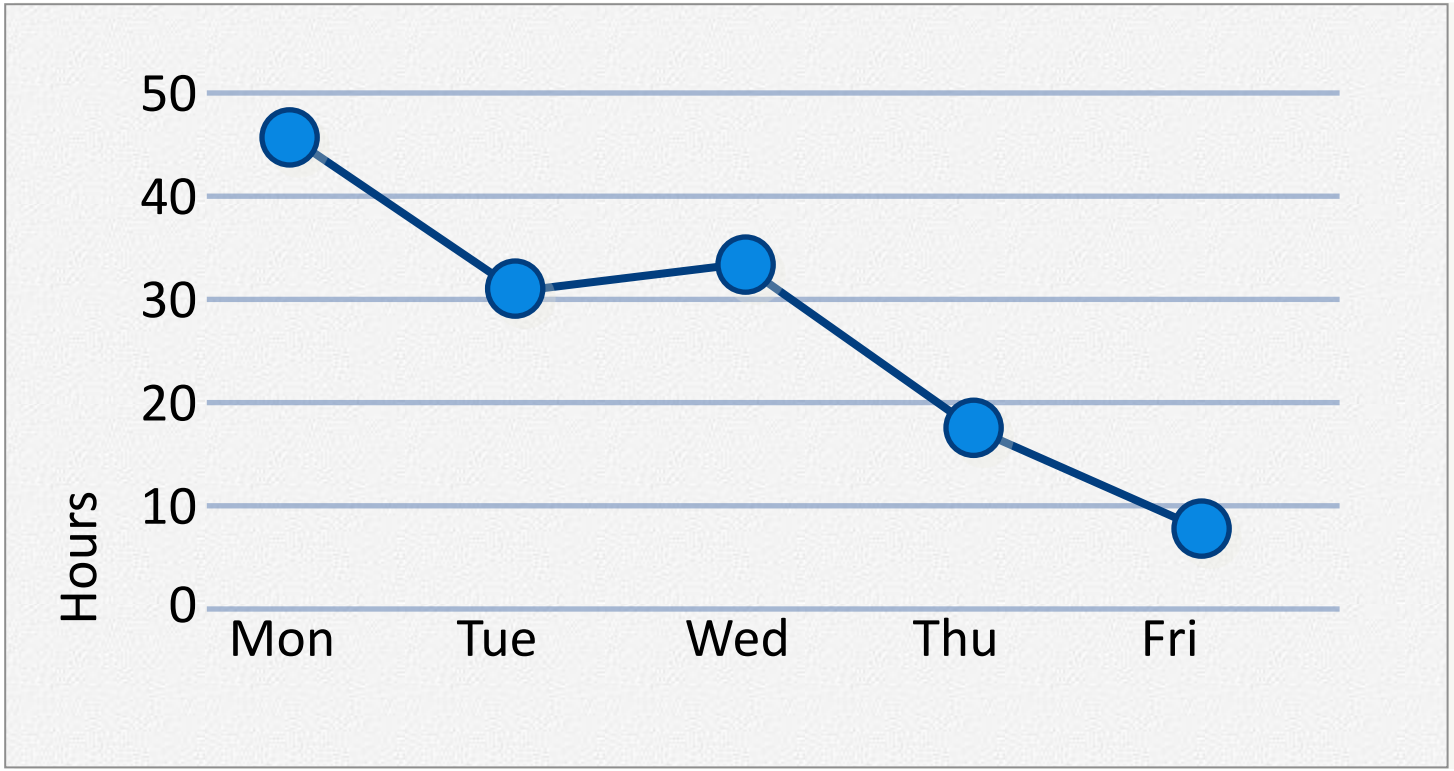
	Backlog Item	Backlog Item	Owner	Estimate	Sprint Day	1 Mo	2 Tu	3 We	4 Th	5 Fr	6 Sa	7 Su
					Hours remaining	152	150	140	130	118	118	118
1	Minor	Remove user kludge in .dpr file	BC	8		8	8	4	2	0		
2	Minor	Remove cMap/cMenu/cMenuSize from disciplines.pas	BC	8		8	8	4	0			
3	Minor	Create "Legacy" discipline node with old civils and E&I content	BC	8		8	8	8	6	0		
4	Major	Augment each tbl operation to support network operation	BC	80		80	80	80	80	78	78	78
5	Major	Extend Engineering Design estimate items to include summaries	BC	16		16	16	16	16	16	16	16
6	Super	Supervision/Guidance	CAM	32		32	30	28	26	24	24	24

Sprint Burndown

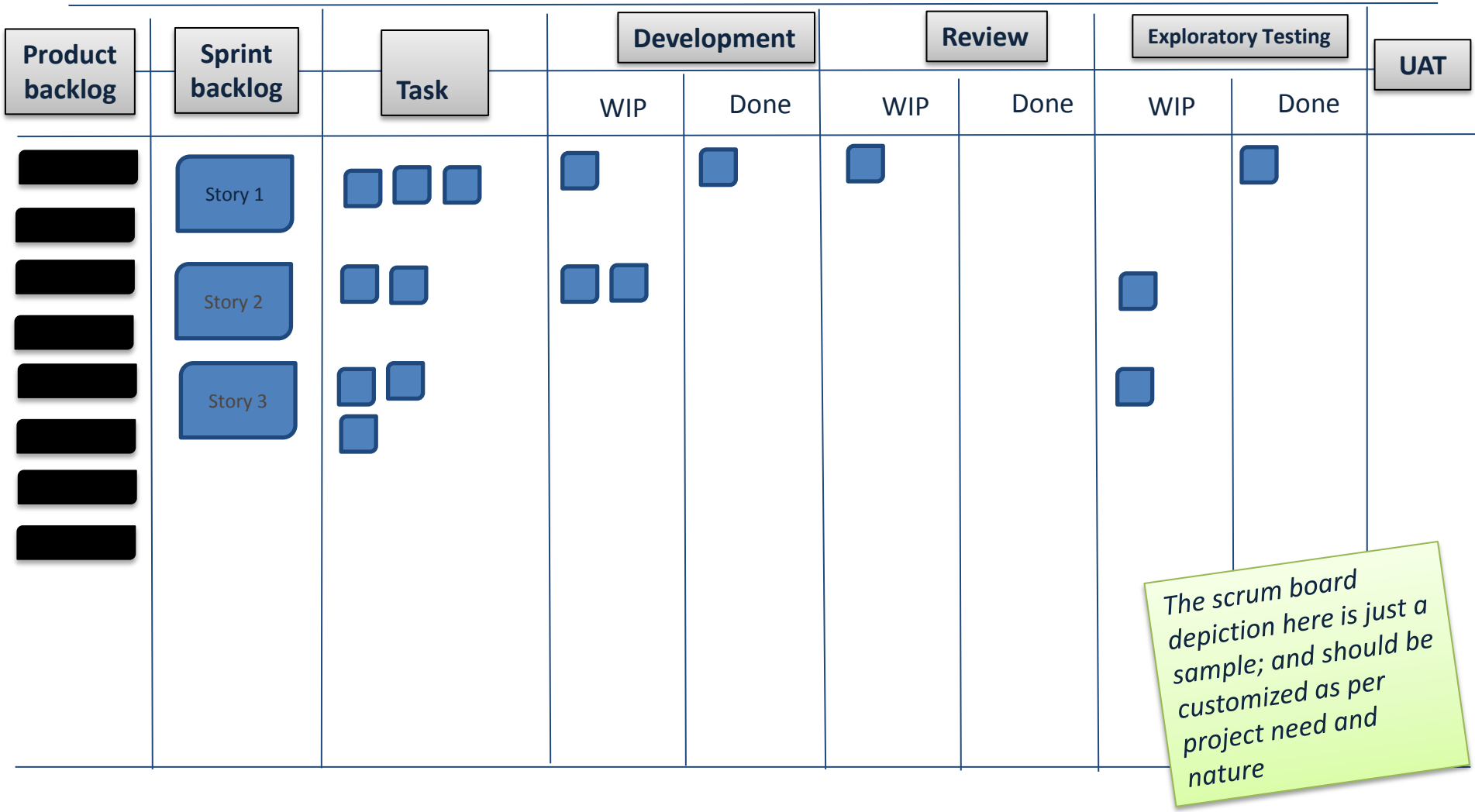
- A display of what work has been completed and what is left to complete
  - one for each developer or work item
  - updated every day
  - (make best guess about hours/points completed each day)
- Gives indication to:
  - No work being performed
  - Not fast enough work
  - Too fast work
- *variation:* Release burndown chart
  - shows overall progress
  - updated at end of each sprint



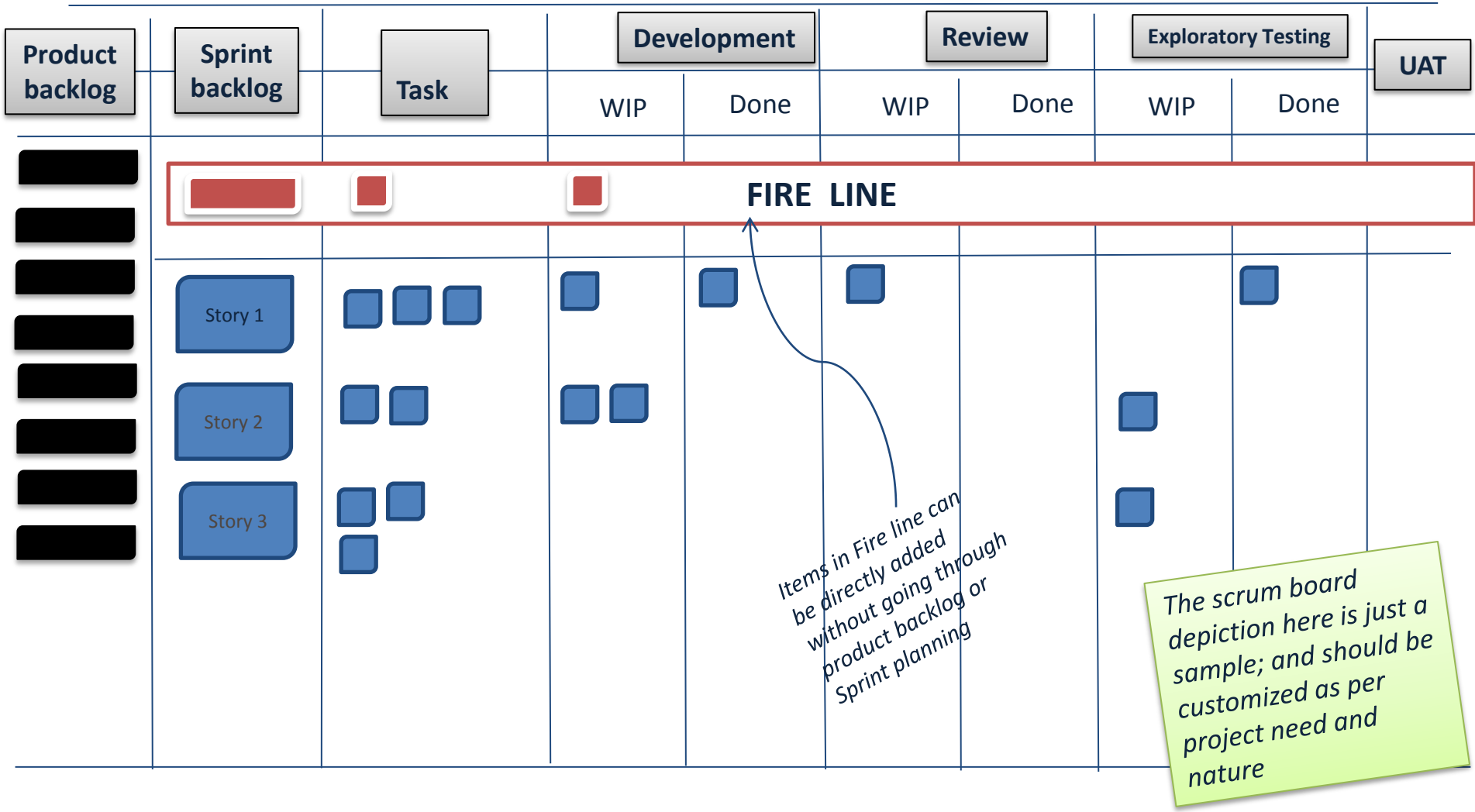
Tasks	Mon	Tues	Wed	Thur	Fri
Code the user interface	8	4	8		
Code the middle tier	16	12	10	7	
Test the middle tier	8	16	16	11	8
Write online help	12				



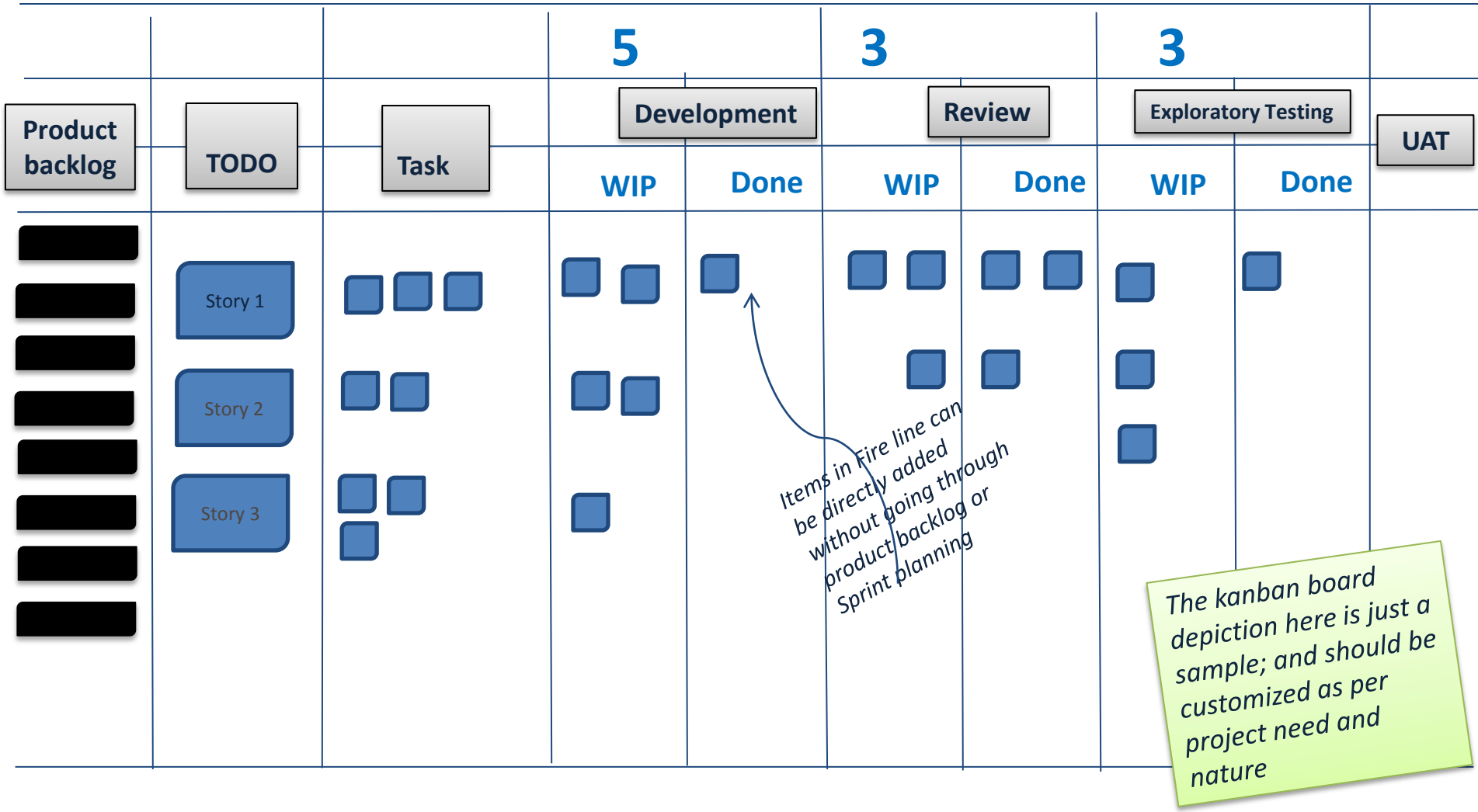
Scrum board


Product backlog
Sprint backlog
Task
Development
WIP
Done
Review
WIP
Done
Exploratory Testing
WIP
Done
UAT
Story 1
Story 2
Story 3
The scrum board depiction here is just a sample; and should be customized as per project need and nature

Fire line on scrum board



Kanban


5
3
3
Product backlog
TODO
Task
Development
Review
Exploratory Testing
UAT
WIP
Done
WIP
Done
WIP
Done
Story 1
Story 2
Story 3
Items in Fire line can be directly added without going through product backlog or Sprint planning
The kanban board depiction here is just a sample; and should be customized as per project need and nature

Differences

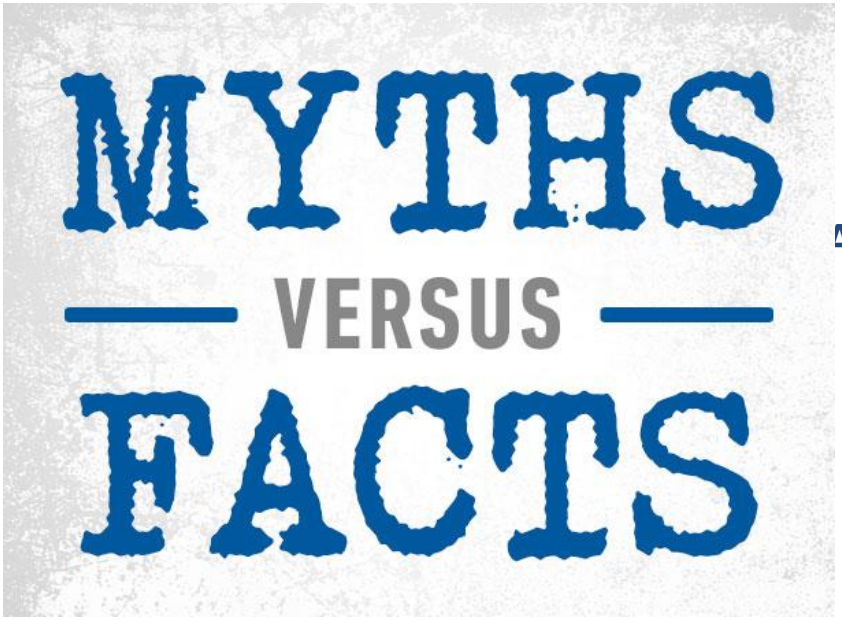
Scrum	Kanban
Timeboxed iterations prescribed.	**Timeboxed iterations optional.**
Team commits to a specific amount of work for this iteration.	**Commitment optional.**
Uses **Velocity** as default metric for planning and process improvement.	Uses **Lead time** as default metric for planning and process improvement.
Cross-functional teams prescribed.	Cross-functional teams optional. **Specialist teams allowed.**
Items broken down so they can be completed within 1 sprint.	No particular item size is prescribed.
Burndown chart prescribed	No particular type of diagram is prescribed
WIP limited indirectly (per sprint)	**WIP limited directly** (per workflow state)
Estimation prescribed	**Estimation optional**
Cannot add items to ongoing iteration.	**Can add new items whenever capacity is available**
A **sprint backlog is owned by one specific team**	A **kanban board may be shared by multiple teams** or individuals
Prescribes 3 roles (PO/SM/Team)	**Doesn't prescribe any roles**
A **Scrum board is reset** between each sprint	A **kanban board is persistent**
Prescribes a prioritized product backlog	**Prioritization is optional.**

Myths or Facts

Agile is undisciplined

Agile is anti documentation

Road map and release plan is essential in Agile



Agile means small water falls

Agile works well for big projects

Agile required lots of rework

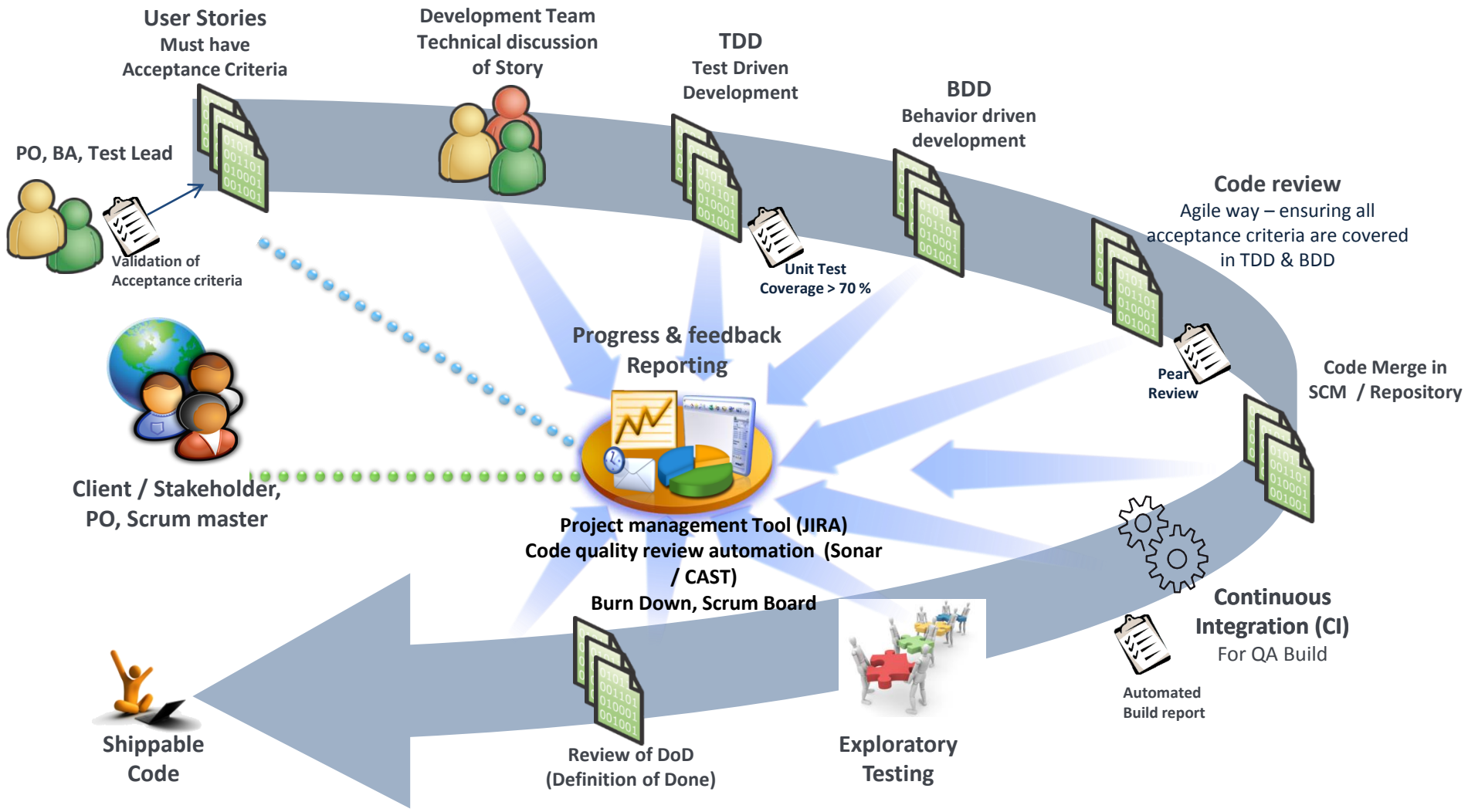
Agile doesn't scale

Agile is anti architecture

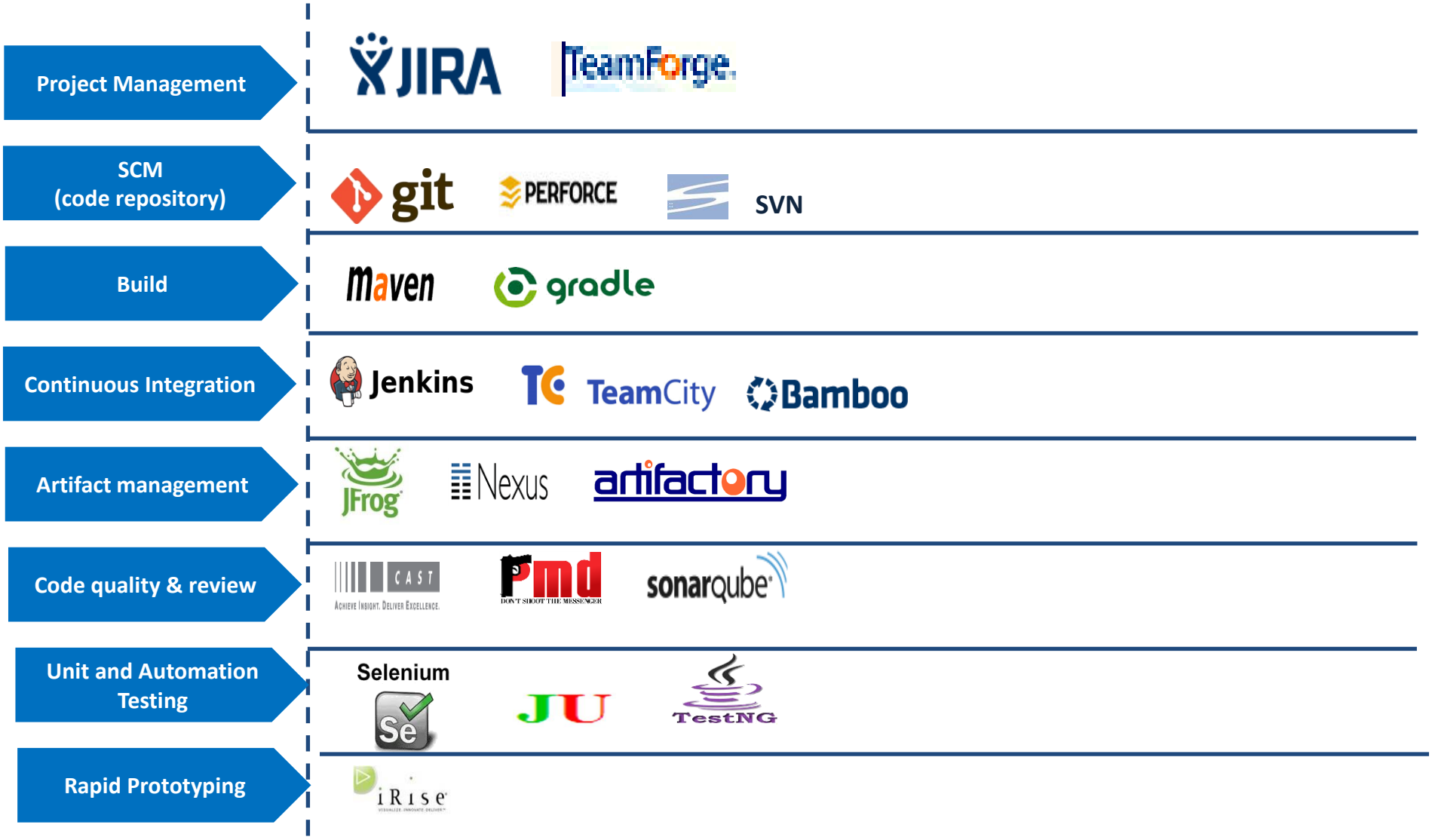
Success Selfie!

- Going agile is all about building a great development culture in your organization.
- Effective cross-functionalism
- Self organizing team
- Don’t be critical-path developers, cross-train each other
- Cross-pollinate though effective code review
- Cultivate full-stack developers

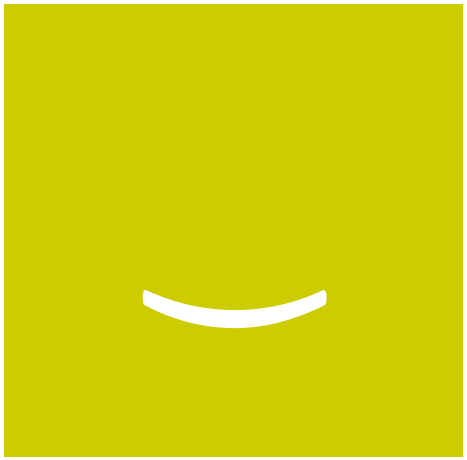
Agile Engineering Practices



Agile Engineering Practices: Suggested Tools



Agile Transformation - Road Map and Milestones



Thank You