Agile Methods

- Agile methods:
  - **Scrum**
  - **Extreme Programming**

Scrum - an agile process

- **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 confusion of conflicting interest and needs
  - Improve communication and maximize cooperation
  - Protecting the team from disturbances
  - A way to maximize productivity

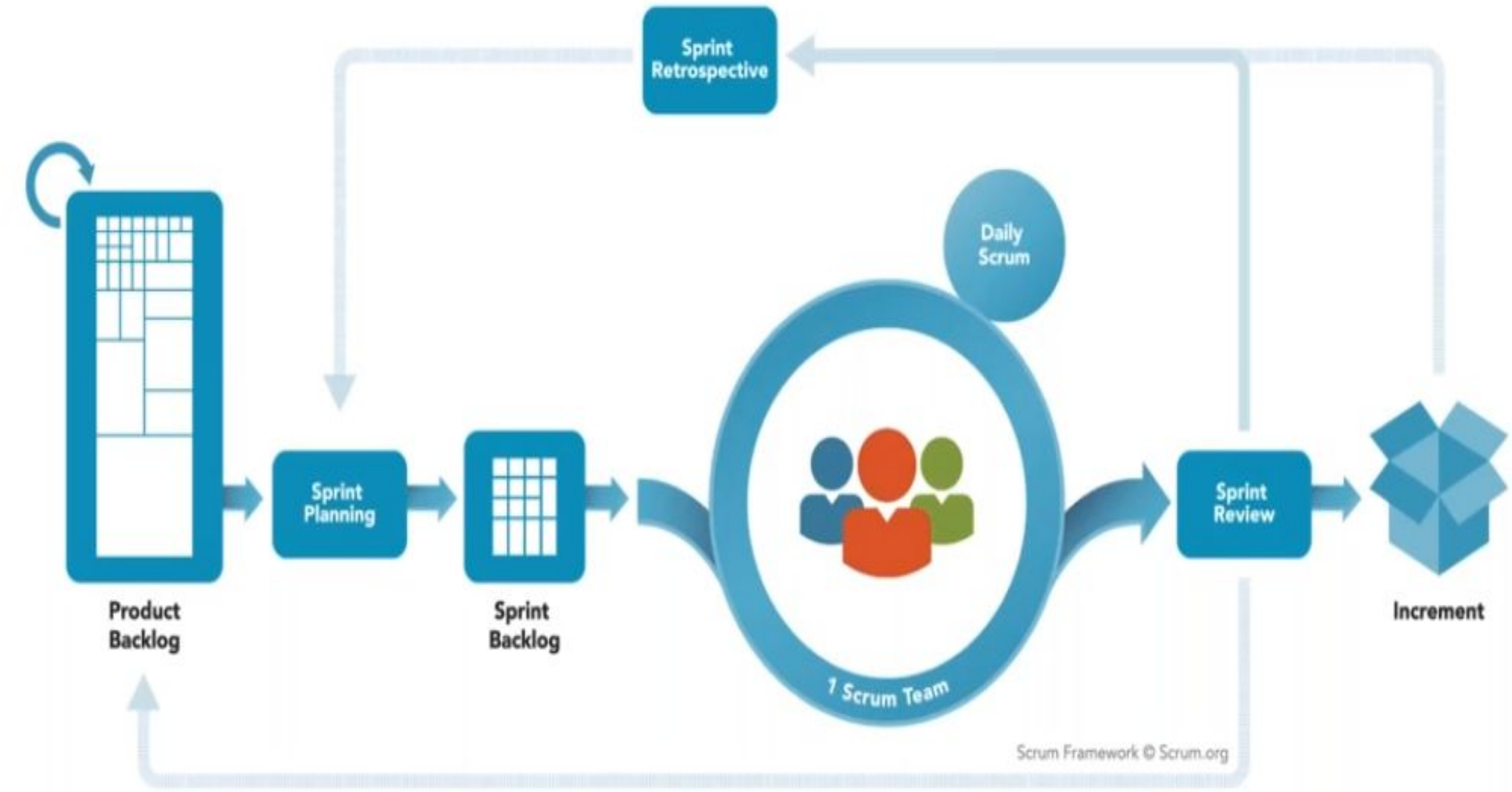
Scrum

- Scrum—distinguishing features
  - Development work is partitioned into “packets”
  - Testing and documentation are on-going as the product is constructed
  - Work units occurs in “sprints” and is derived from a “backlog” of existing changing prioritized requirements
  - Changes are not introduced in sprints (short term but stable) but in backlog.
  - Meetings are very short (15 minutes daily) and sometimes conducted without chairs
    - What did you do since last meeting?
    - What obstacles are you encountering?
    - What do you plan to accomplish by next meeting?
  - “Demos” are delivered to the customer with the time-box allocated. May not contain all functionalities. So customers can evaluate and give feedbacks.

Characteristics

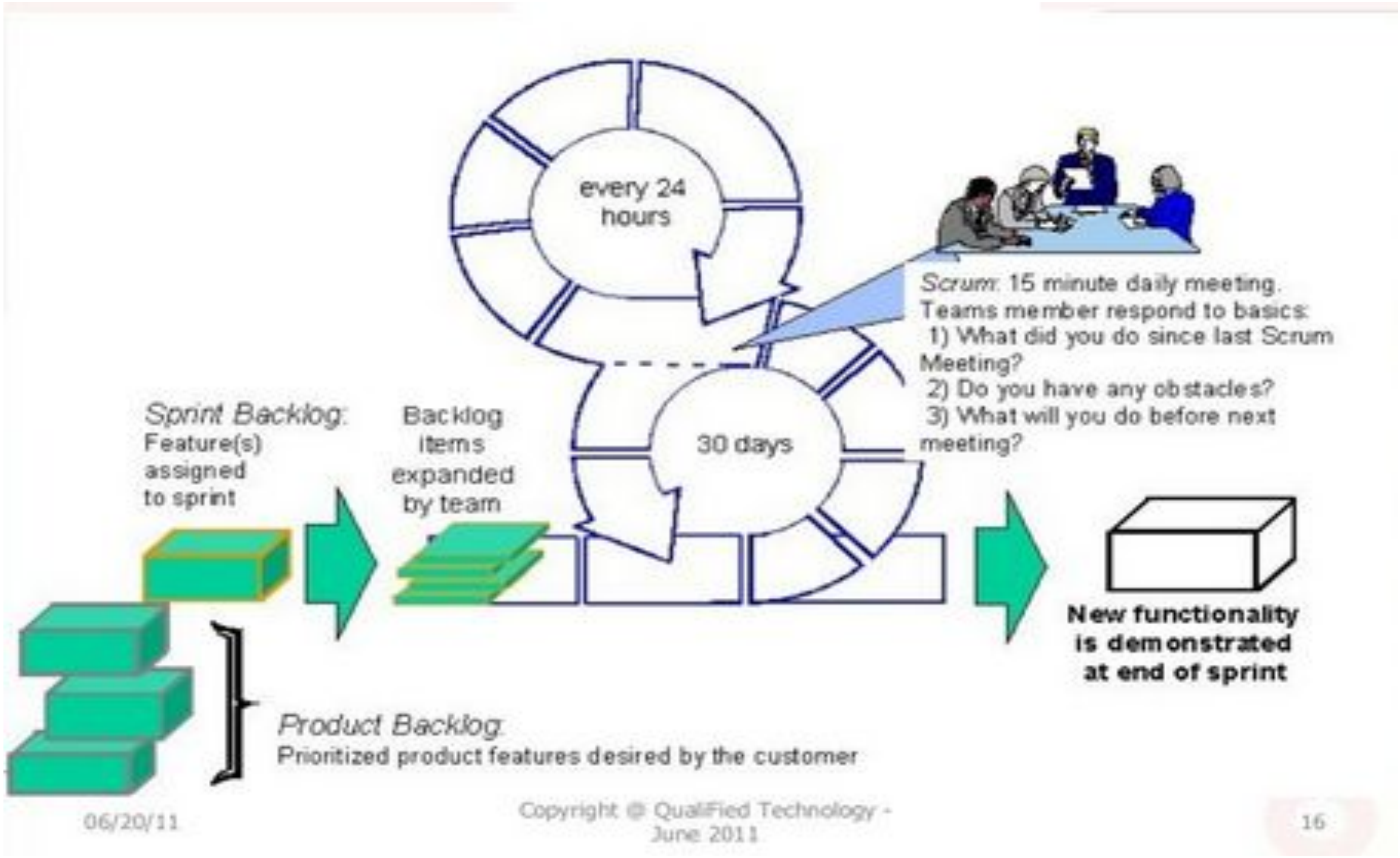
- Self-organizing teams.
- Product progresses in a series of month-long “sprints”.
- Requirements are captured as items in a list of “product backlog”
- No specific engineering practices prescribed.
- Uses generative rules to create an agile environment for delivering projects.
- One of the “agile processes”.

SCRUM



Scrum Framework © Scrum.org

Scrum Process Flow


every 24 hours
30 days
Backlog items expanded by team
New functionality is demonstrated at end of sprint

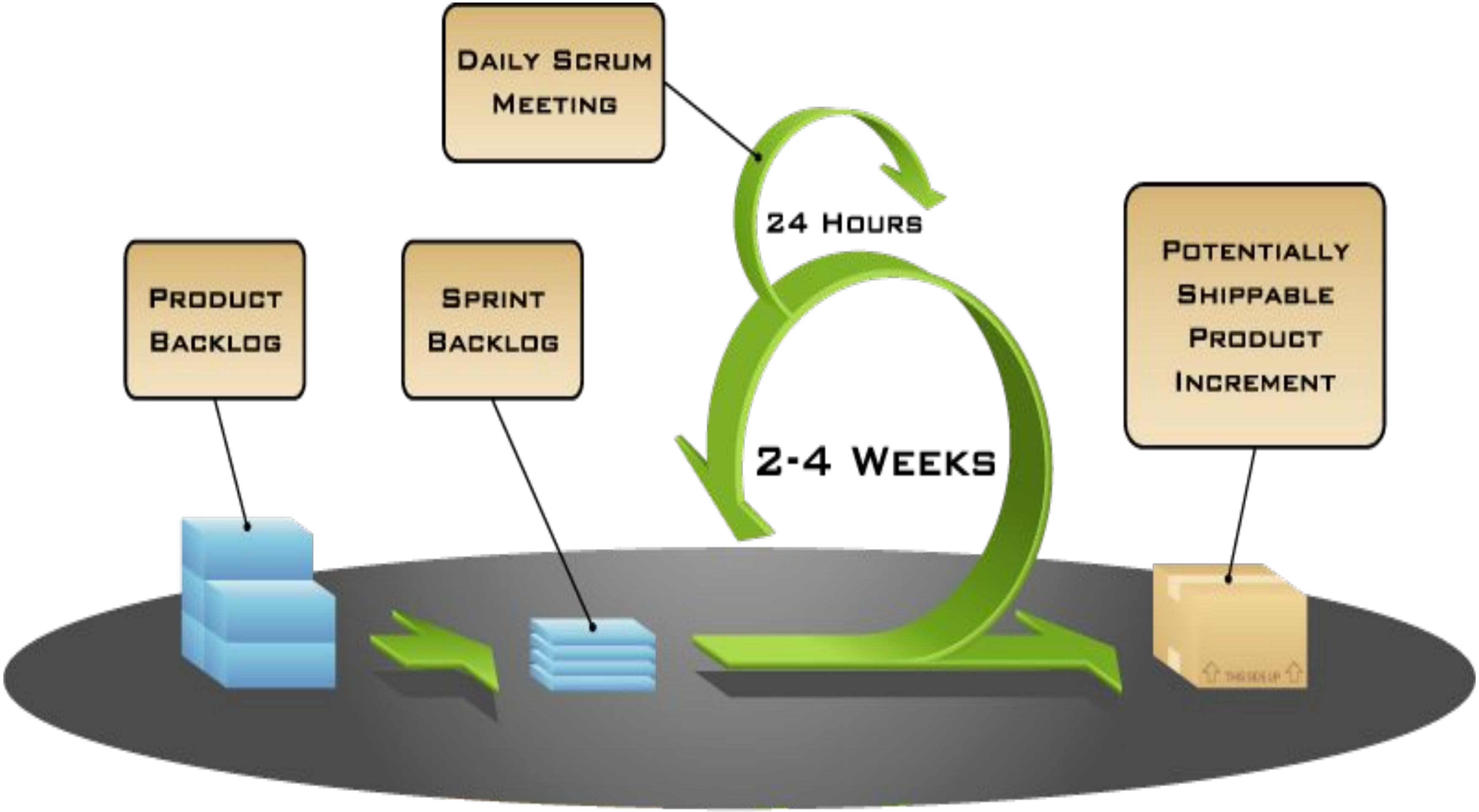
Sprint Backlog: Feature(s) assigned to sprint

Scrum: 15 minute daily meeting.
Teams member respond to basics:
1) What did you do since last Scrum Meeting?
2) Do you have any obstacles?
3) What will you do before next meeting?

Product Backlog: Prioritized product features desired by the customer

06/20/11 Copyright @ QualiFied Technology - June 2011 16

Functionality of Scrum



Copyright © 2005, Mountain Goat Software

Project Management Emphasis based on a Standard 30-day Sprint

- **Scrum:** a definite project management emphasis.

- **Scrum Master**: A **Scrum project** is managed by a Scrum Master, who can be considered as much a consultant or coach as a manager.

- **Sprint**. Scrum has a fundamental 30-day development cycle called a **Sprint**, preceded by
  - **pre-Sprint** activities and **post-Sprint** activities.

- **Daily Scrum**: A short (less than 30 minutes) daily Scrum Meeting allows the team to monitor status and communicate problems.

Product Backlog for Planning

- **Project planning** is based on a **Product Backlog**, which contains
  - functions and
  - technology enhancements
- Intended for the project.

- Two meetings are held –
  - One to decide the **features for the next Sprint** and
  - The other to **plan out the work**.

Scrum - Queues

- Product Backlog ⍰ Sprint Backlog ⍰ Sprint ⍰ Working increment of the Software
- **Scrum** uses **lightweight queue-based management** and work-breakdown mechanisms.
- **Product Backlog queue**: a low-tech customer-managed queue of demand requests for products.
- **Sprint**: At launch time, a Sprint (30-day time-boxed iteration) does **just-in-time planning**
- **Sprint Backlog**: queue for Sprint work-mgmt.

Scrum - Management

- **Daily Scrum:** Very notable and very visible
- Is a **daily standup**,
  - **except** that it is the **team** that is participating and sharing coordination information **not** a **central project manager**.

- **Scrum Master**
  - holds daily scrum and
  - acts more as a **facilitator** and **runs interference** for the core team when **blocks** or **issues** arise. (Kennaley, SDLC 3.0, p. 31)

Components of Scrum

- Scrum Roles
- The Process
- Scrum Artifacts

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.
- Knows what needs to be build and in what sequence this should be done
- Typically a product manager

Core Roles – **Product Owner**

- The Product Owner represents **stakeholders** and is the **voice of the customer**.
- Product Owner is **accountable** for ensuring that the team **delivers value** to the business.
- **Product Owner**
  - **writes** customer-centric items (typically **user stories**),
  - **prioritizes** them, and
  - **adds** them to the **product backlog**.

Note:

- Scrum teams should have **one Product Owner**.
- May also be a member of the development team
- Not recommend this person be Scrum Master.

Core Roles – Scrum Master

- **Scrum** is facilitated by a Scrum Master –
- Accountable for **removing impediments** for team to deliver sprint goal / deliverables.
- **Scrum Master is not the team leader,** but acts as a **buffer** between the team and any distracting influences.
- Scrum Master ensures **process** is used as intended.
- Scrum Master is the **enforcer of rules**.
- Scrum Master's role: **protect** the Team and keep it **focused** on the tasks at hand.

The Scrum Master

- Represents management to the project
- 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

Core Roles – **Development Team**

- The Development Team is responsible for **delivering potentially shippable product increments** at end of each Sprint.
- Cross-functional
  - QA, Programmers, UI Designers, etc.
- Team = 3–9 people with cross-functional skills.
- Team does actual work
  - (analyze, design, develop, test, technical communication, document, etc.).
- Team is **self-organizing**, even though they may interface with project management organizations (PMOs).
- Membership can change only between sprints

The Process

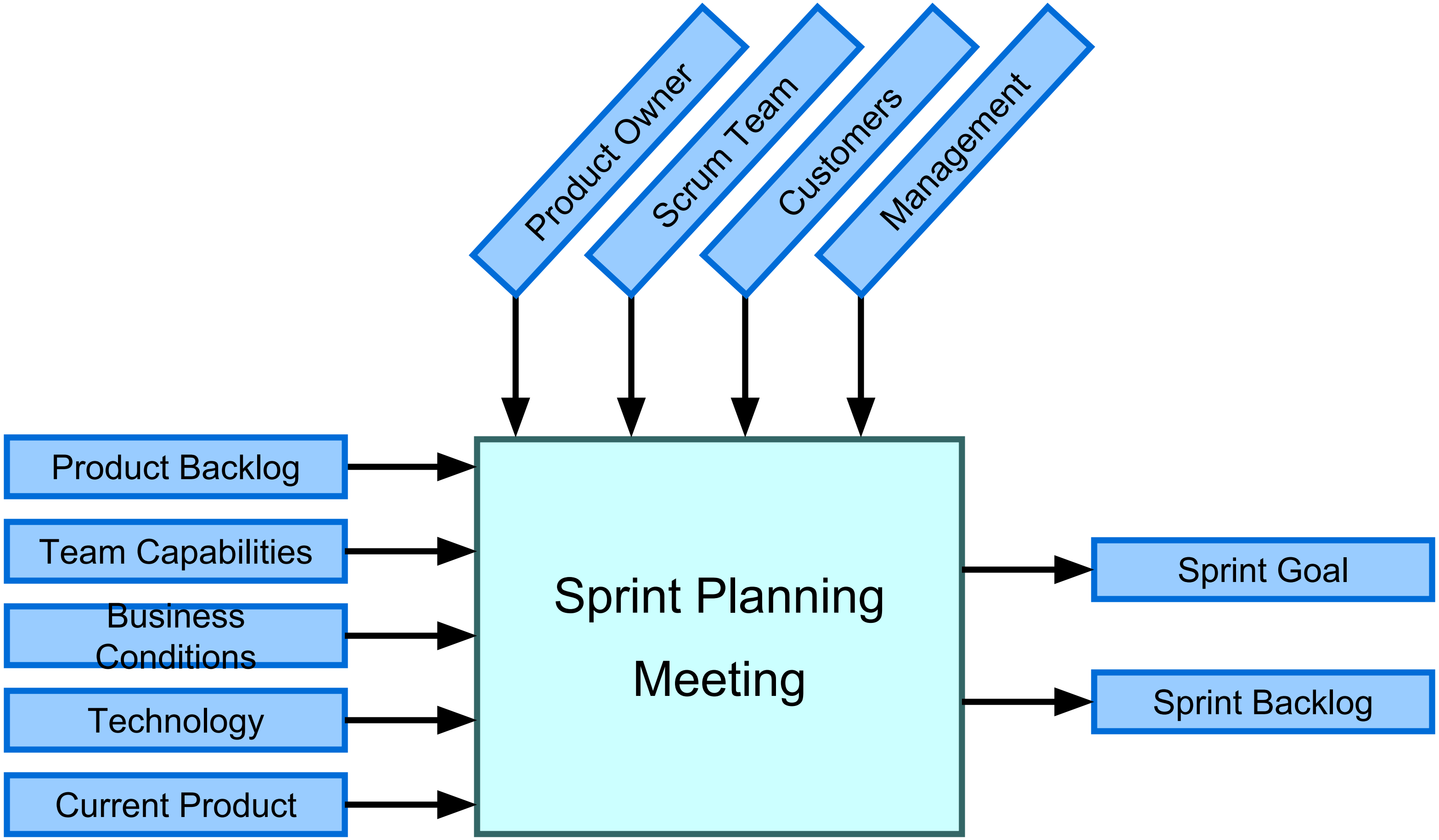
- Sprint Planning Meeting
- Sprint
- Daily Scrum
- Sprint Review Meeting

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”)

Sprint Planning Meeting



Parts of Sprint Planning Meeting

- 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.

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

Sprints

- Scrum projects make progress in a series of “sprints”
  - Analogous to XP iterations.
- Target duration is one month
  - +/- a week or two
    - But, a constant duration leads to a better rhythm
- Product is designed, coded, and tested during the sprint.

- **Sprint**: basic unit of development in Scrum.
- **Sprint duration**: one week to one month;
- **“Time Boxed"** effort of a constant length.

- Each sprint:
- Preceded by a **planning meeting**,
  - where the **tasks** for sprint are **identified** and an **estimated commitment for the sprint goal** made, and followed by a **review or retrospective meeting**, where the progress is reviewed and lessons for the next sprint are identified.

- During Sprint, **team creates finished portions** of a product. (an **increment**)

- **Features** going into a Sprint come from the ***product backlog***: a prioritized list of requirements.
  - **Which backlog items** go into sprint (**sprint goals)** determined during Sprint Planning Mtg.

- **Sprint Goal**
  - sets up **minimum success criterion** for the Sprint and
  - **keeps the team focused** on the broader picture rather than narrowly on the task at hand.

- The **team then** determines **how many selected items** can be **completed** during the next sprint.

- These then go into the **Sprint Backlog**.

- **Sprint Backlog** is **property** of the development team, During a sprint, **no one is allowed to edit the sprint backlog except for development team.**

- **Development: time-boxed**; Sprint **must** end on time;

- Requirements not completed for any reason?
  are omitted and **returned** to **Product Backlog**.

- When Sprint is done, team **demonstrates** software.

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

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

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?

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

Sprint Retrospective Meeting

- Scrum Team only
- Feedback meeting
- Three questions
  - Start
  - Stop
  - Continue
- Don’t skip for the first 5-6 sprints!!!

Scrum Artifacts

- Product Backlog
- Sprint Backlog
- Burn down Charts

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 Planning Meeting

Product Backlog

- A list of all desired work on the project
  - Usually a combination of
    - story-based work (“let user search and replace”)
    - task-based work (“improve exception handling”)
- List is prioritized by the Product Owner
  - Typically a Product Manager, Marketing, Internal Customer, etc.

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

Artifact: Product Backlog

- **Product backlog** is an ordered list of "requirements" that is maintained for a product

- Contains Product Backlog Items **ordered** by the Product Owner based on
  - considerations like risk,
  - business value,
  - dependencies,
  - date needed, etc.

- **Features** added to backlog commonly written in story format

- The product backlog is the “**What**” that will be built, sorted in the relative order it should be built in.
  - **Product Owner is ultimately responsible** for **ordering the stories** on the backlog for the Development Team.

Artifact: Product Backlog

- The product backlog contains rough estimates of both business value and development effort, these values are often stated in **story points** using a rounded Fibonacci sequence.

- Those **estimates** help the Product Owner to gauge the timeline and may influence ordering of backlog items.

  – Example, if the “add spellcheck” and “add table support” features have the same business value, the one with the smallest development effort will probably have higher priority, because the Return on Investment is higher.

Artifacts – The Product Backlog 2

- **Product Owner**: responsible for **the product backlog** and the **business value** of each item listed.
- **Development Team**: responsible for the **estimated effort** to complete each backlog item.

- Team contributes by estimating Items and User-Stories, either in "**Story-points**" or in "**estimated hours**."

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

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.
- Is a good warning monitor

From Sprint Goal to Sprint Backlog

- Scrum team takes the Sprint Goal and decides what tasks are necessary
- Team self-organizes around how they'll meet the Sprint Goal
  - Manager doesn't assign tasks to individuals
- Managers don't make decisions for the team
- Sprint Backlog is created

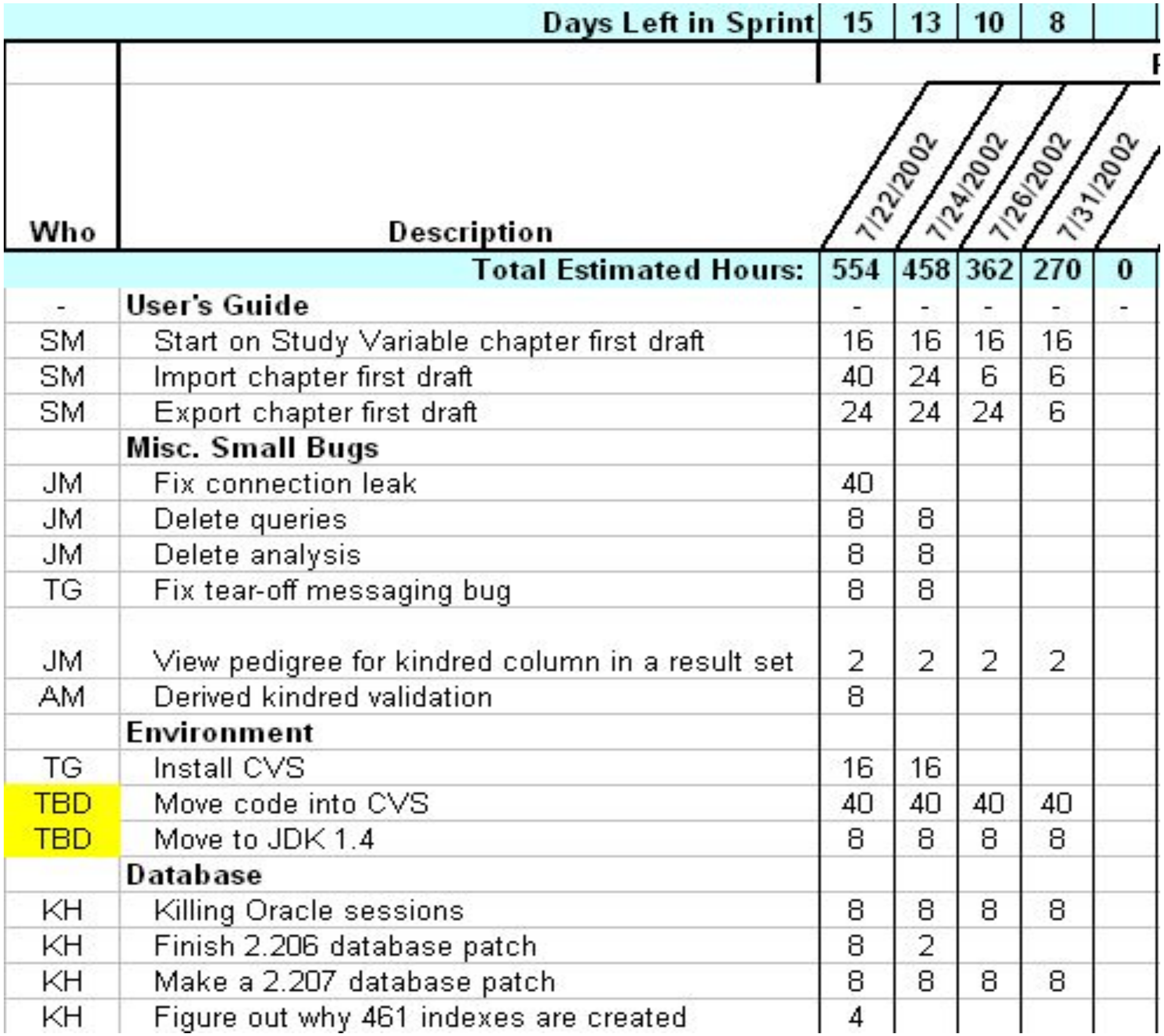
Sprint Backlog during the Sprint

- Changes
  - Team adds new tasks whenever they need to in order to meet the Sprint Goal
  - Team can remove unnecessary tasks
  - But: Sprint Backlog can only be updated by the team
- Estimates are updated whenever there's new information

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

Sample Sprint Backlog



	Days Left in Sprint	15	13	10	8	
Who	Description	7/22/2002	7/24/2002	7/26/2002	7/31/2002	
	Total Estimated Hours:	554	458	362	270	0
-	**User's Guide**	-	-	-	-	-
SM	Start on Study Variable chapter first draft	16	16	16	16	
SM	Import chapter first draft	40	24	6	6	
SM	Export chapter first draft	24	24	24	6	
	Misc. Small Bugs					
JM	Fix connection leak	40				
JM	Delete queries	8	8			
JM	Delete analysis	8	8			
TG	Fix tear-off messaging bug	8	8			
JM	View pedigree for kindred column in a result set	2	2	2	2	
AM	Derived kindred validation	8				
	Environment					
TG	Install CVS	16	16			
TBD	Move code into CVS	40	40	40	40	
TBD	Move to JDK 1.4	8	8	8	8	
	Database					
KH	Killing Oracle sessions	8	8	8	8	
KH	Finish 2.206 database patch	8	2			
KH	Make a 2.207 database patch	8	8	8	8	
KH	Figure out why 461 indexes are created	4				

Artifacts: Sprint Backlog

- **Sprint Backlog:** list of work the Development Team must address during the next sprint.
- List derived by selecting stories/features from the top of the product backlog until the Development Team feels it has enough work to fill the sprint.

- **Thinking**: This is done by the Development Team asking "Can we also do this?" and adding stories/features to the sprint backlog.

- **History**: Development Team should note **velocity** of previous Sprints (total story points completed from each of the last sprints stories) when selecting stories/features for the **new sprint.**

- Use number as **guide** for "effort" they can complete.

Artifacts: Sprint Backlog

- **Stories/features**: broken down into **tasks** by Development Team- should normally be between **four and sixteen hours** of work.

- With this level of detail the Development Team understands exactly what to do, and potentially, anyone can pick a task from the list.

- **Tasks** on sprint backlog are **never assigned**; tasks are **signed up for.** by team members during **daily scrum**, according to **priority** and **member skills**.

- **Sprint backlog is property of Team**, and all included **estimates** are provided by the Development Team.

Artifacts - Increment

- The "**increment**" is **sum** of all Product Backlog Items completed during a sprint **and** all previous sprints.
- At end of a sprint, Increment must be done according to **Scrum Team's definition** of done.
- The increment must be in **usable condition** regardless of whether the Product Owner decides to actually release it.

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)

Artifacts: Burn Down

- The sprint **burn down chart** is a publicly displayed chart showing **remaining work** in the sprint backlog.

- Updated every day; gives a simple view of the sprint progress.

- Other types of burn down:
- **Release burn down chart:** shows amount of work left to **complete** the target commitment for a Product Release
  - This normally spans multiple iterations

- **Alternative Release burn down chart**: basically does the same, but clearly shows scope changes to Release Content, by resetting the baseline.

Sprint Burn down Chart

- Depicts the total Sprint Backlog hours remaining per day
- Shows the estimated amount of time to release
- Ideally should burn down to zero to the end of the Sprint.

Release Burn down Chart

- Will the release be done on right time?
- X-axis: sprints
- Y-axis: amount of hours remaining
- The estimated work remaining can also burn up

Alternative Release Burn down Chart

- Consists of bars (one for each sprint)
- Values on the Y-axis: positive AND negative
- Is more informative then a simple chart

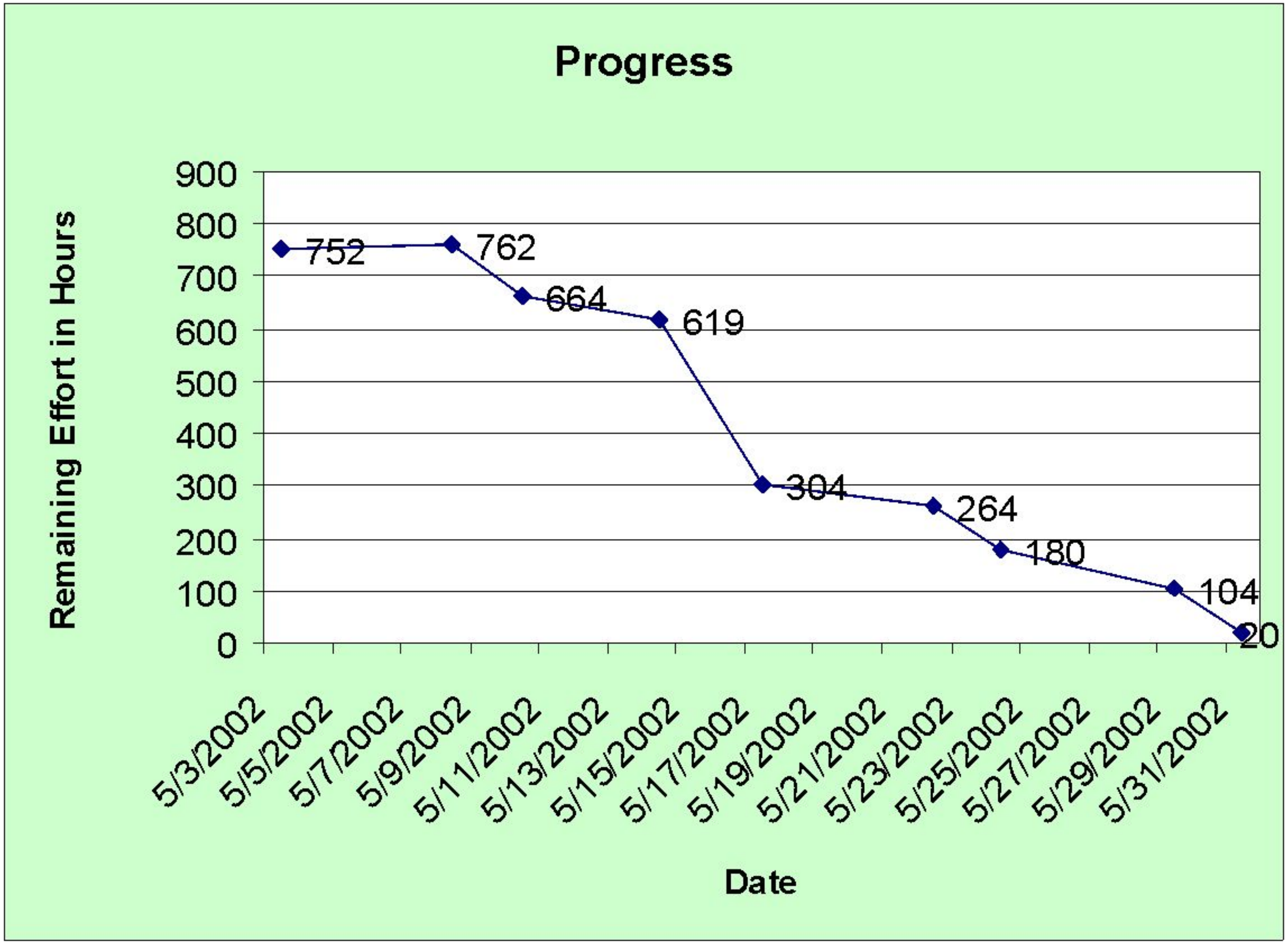
Product Burn down Chart

- Is a “big picture” view of project’s progress (all the releases)

Scaling Scrum

- A typical Scrum team is 6-10 people
- Jeff Sutherland - up to over 800 people
- "Scrum of Scrums" or what called "Meta-Scrum“
- Frequency of meetings is based on the degree of coupling between packets

Sprint Burndown Chart



XP@Scrum

Scrum is an effective project management wrapper for eXtreme Programming development practices, which enables agile projects to become scalable and developed by distributed teams of developers.

Pros/Cons

- *Advantages*
  - *Completely developed and tested features in short iterations*
  - *Simplicity of the process*
  - *Clearly defined rules*
  - *Increasing productivity*
  - *Self-organizing*
  - *each team member carries a lot of responsibility*
  - *Improved communication*
  - *Combination with Extreme Programming*

- *Drawbacks*
  - *“Undisciplined hacking” (no written documentation)*
  - *Violation of responsibility*
  - *Current mainly carried by the inventors*

Roles

PO

Product Owner: Set priorities

SM

ScrumMaster: Manage process, remove blocks

T

Team: Develop product

SH

Stakeholders: observe & advise

Key Artifacts

Product Backlog

- List of requirements & issues
- Owned by Product Owner
- Anybody can add to it

Sprint Goal

- One-sentence summary
- Declared by Product Owner

Sprint Backlog

- List of tasks
- Owned by team

Blocks List

- List of blocks & unmade decisions
- Owned by ScrumMaster

Increment

- Version of the product
- Shippable functionality (tested,

Key Meetings

Sprint Planning Meeting

- Hosted by ScrumMaster; ½-1 day
- In: Product Backlog, existing product, business & technology conditions

1. Select highest priority items in Product Backlog; declare Sprint Goal

2. Team turns selected items into

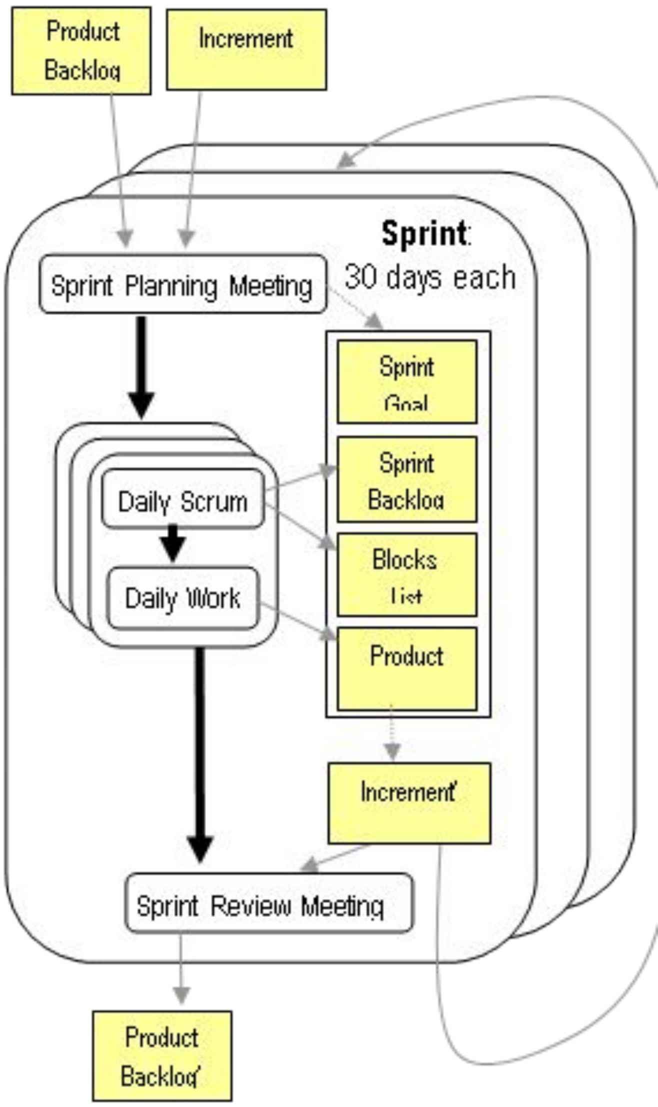
Daily Scrum

- Hosted by ScrumMaster
- Attended by all, but Stakeholders don't speak
- Same time every day
- Answer: 1) What did you do yesterday? 2) What will you do today? 3) What's in your way?
- Team updates Sprint Backlog;

Sprint Review Meeting

- Hosted by ScrumMaster
- Attended by all
- Informal, 4-hour, informational
- Team demos Increment
- All discuss
- Hold retrospective
- Announce next Sprint Planning

Development Process

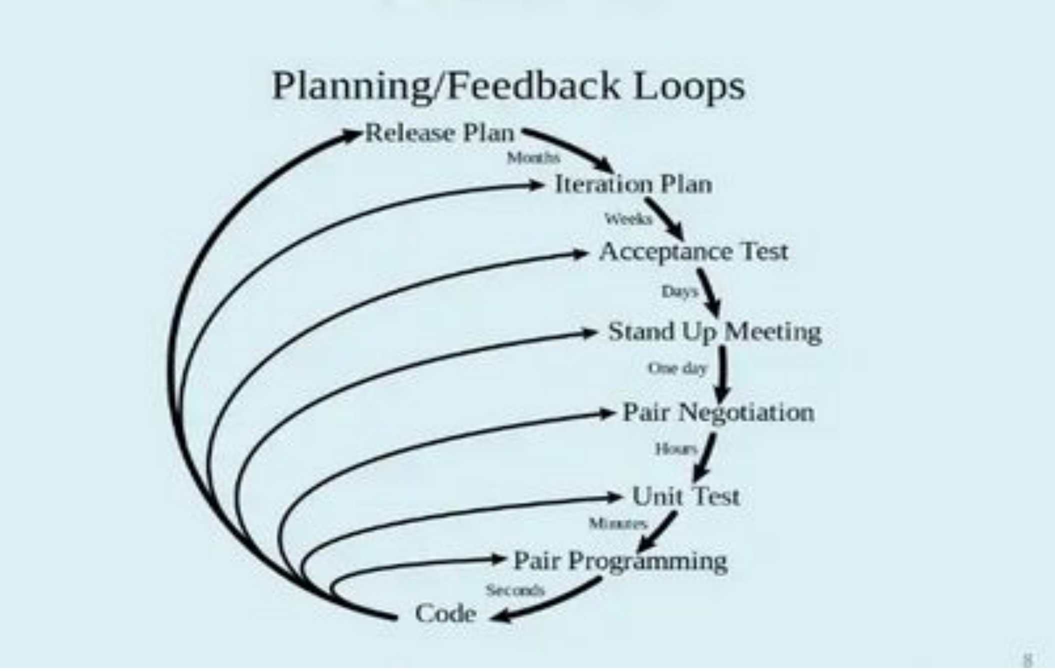


Extreme Programming

Ref : https://www.slideshare.net/DamianGordon1/the-extreme-programming-xp-model

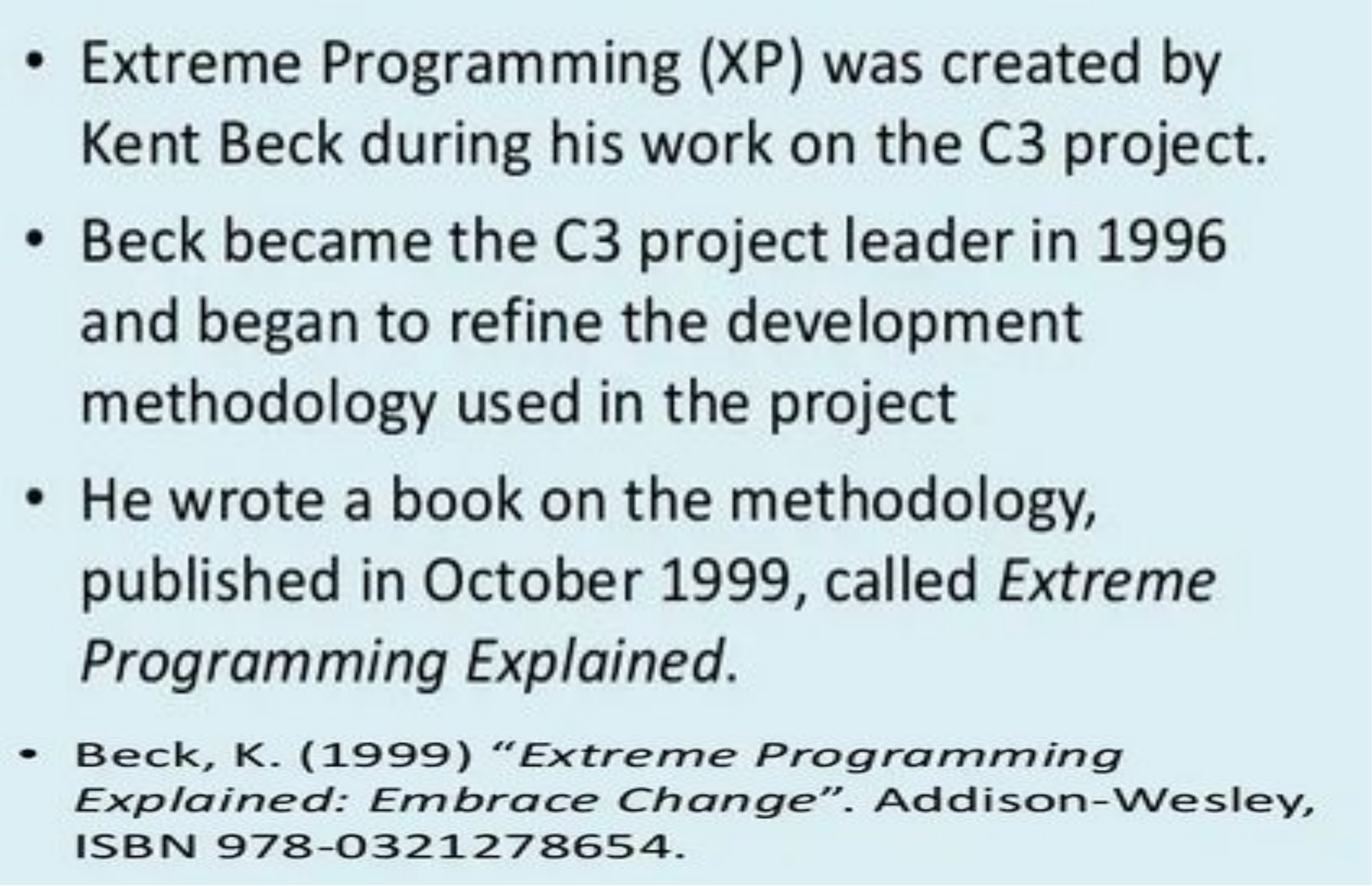
Overview

- The “Extreme Programming XP model” is a model that represents one method as to how software can be developed



REF : https://www.slideshare.net/DamianGordon1/the-extreme-programming-xp-model

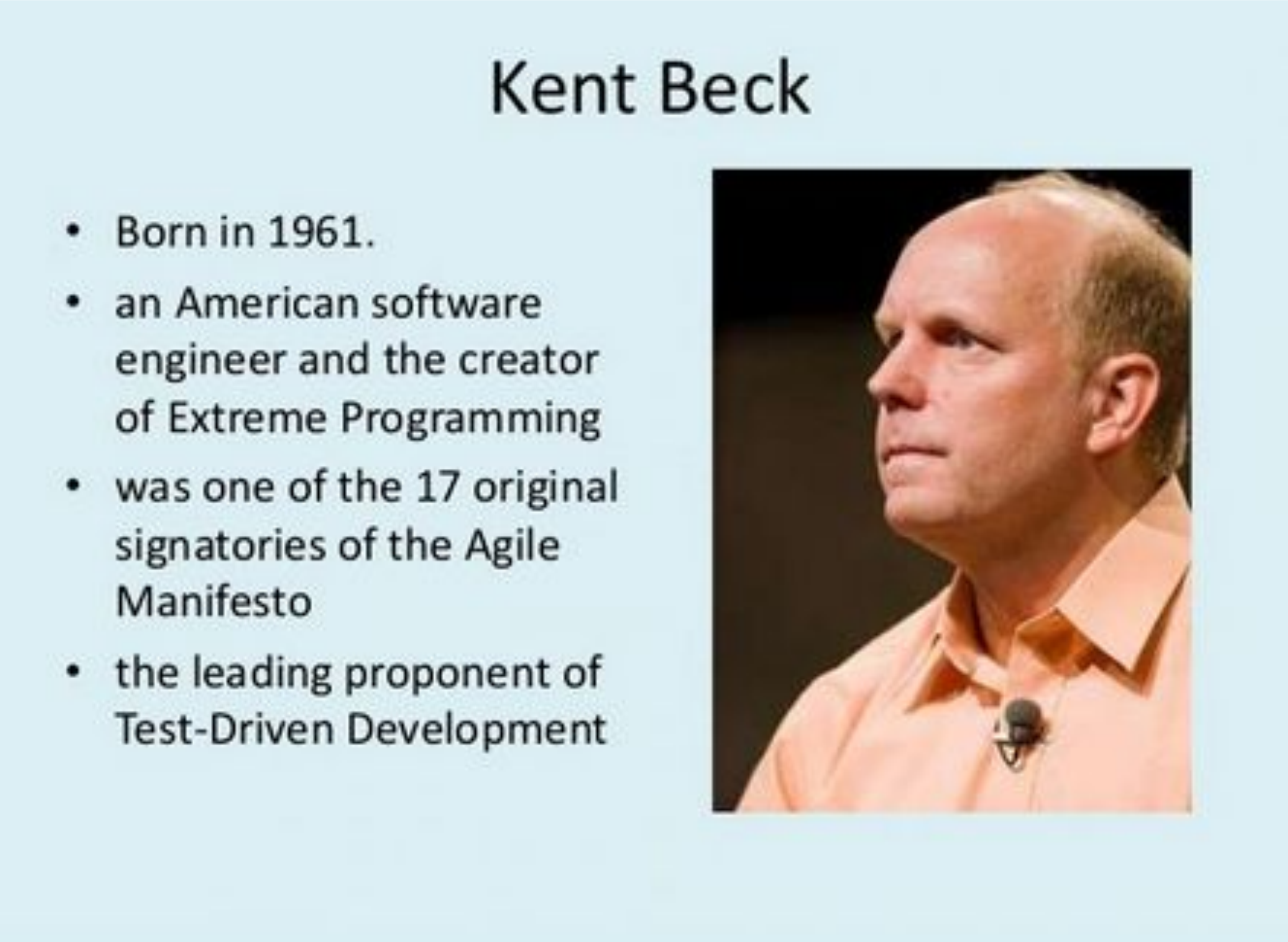
Overview



- Extreme Programming (XP) was created by Kent Beck during his work on the C3 project.
- Beck became the C3 project leader in 1996 and began to refine the development methodology used in the project
- He wrote a book on the methodology, published in October 1999, called *Extreme Programming Explained*.

- Beck, K. (1999) “*Extreme Programming Explained: Embrace Change*”. Addison-Wesley, ISBN 978-0321278654.

REF : https://www.slideshare.net/DamianGordon1/the-extreme-programming-xp-model

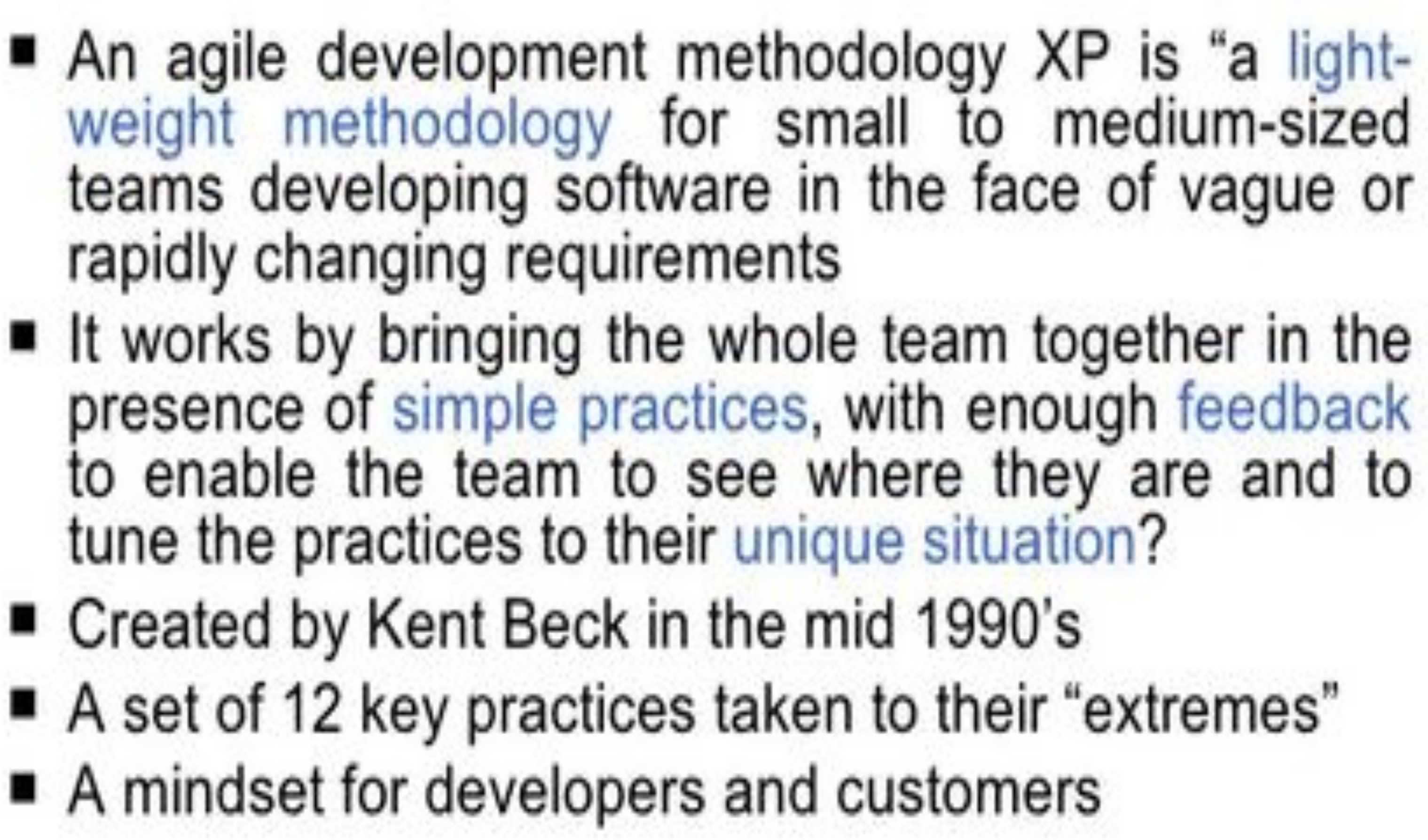


Kent Beck

- Born in 1961.
- an American software engineer and the creator of Extreme Programming
- was one of the 17 original signatories of the Agile Manifesto
- the leading proponent of Test-Driven Development

REF : https://www.slideshare.net/DamianGordon1/the-extreme-programming-xp-model

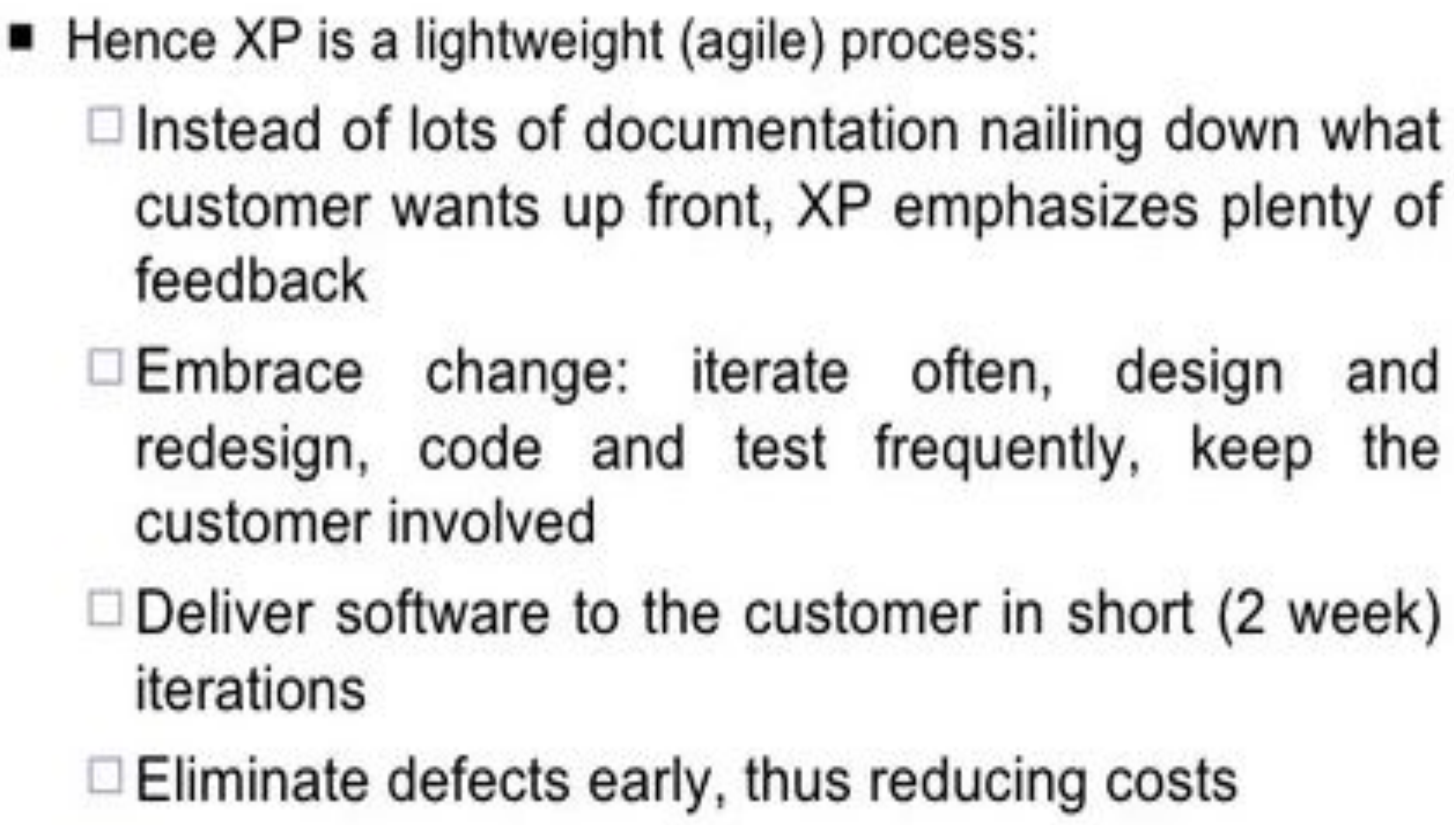
Introduction to Extreme programming



- An agile development methodology XP is "a light-weight methodology for small to medium-sized teams developing software in the face of vague or rapidly changing requirements
- It works by bringing the whole team together in the presence of simple practices, with enough feedback to enable the team to see where they are and to tune the practices to their unique situation?
- Created by Kent Beck in the mid 1990's
- A set of 12 key practices taken to their "extremes"
- A mindset for developers and customers

REF : https://www.slideshare.net/bilalhashmishah/extreme-programming-12836919

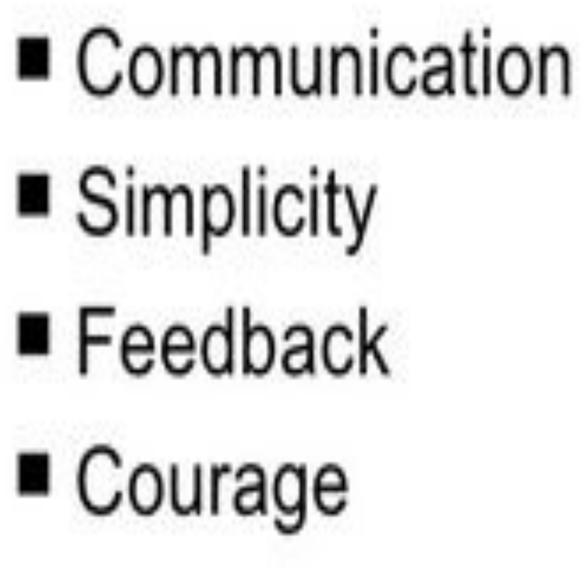
Extreme Programming



- Hence XP is a lightweight (agile) process:
  - □ Instead of lots of documentation nailing down what customer wants up front, XP emphasizes plenty of feedback
  - □ Embrace change: iterate often, design and redesign, code and test frequently, keep the customer involved
  - □ Deliver software to the customer in short (2 week) iterations
  - □ Eliminate defects early, thus reducing costs

REF : https://www.slideshare.net/bilalhashmishah/extreme-programming-12836919

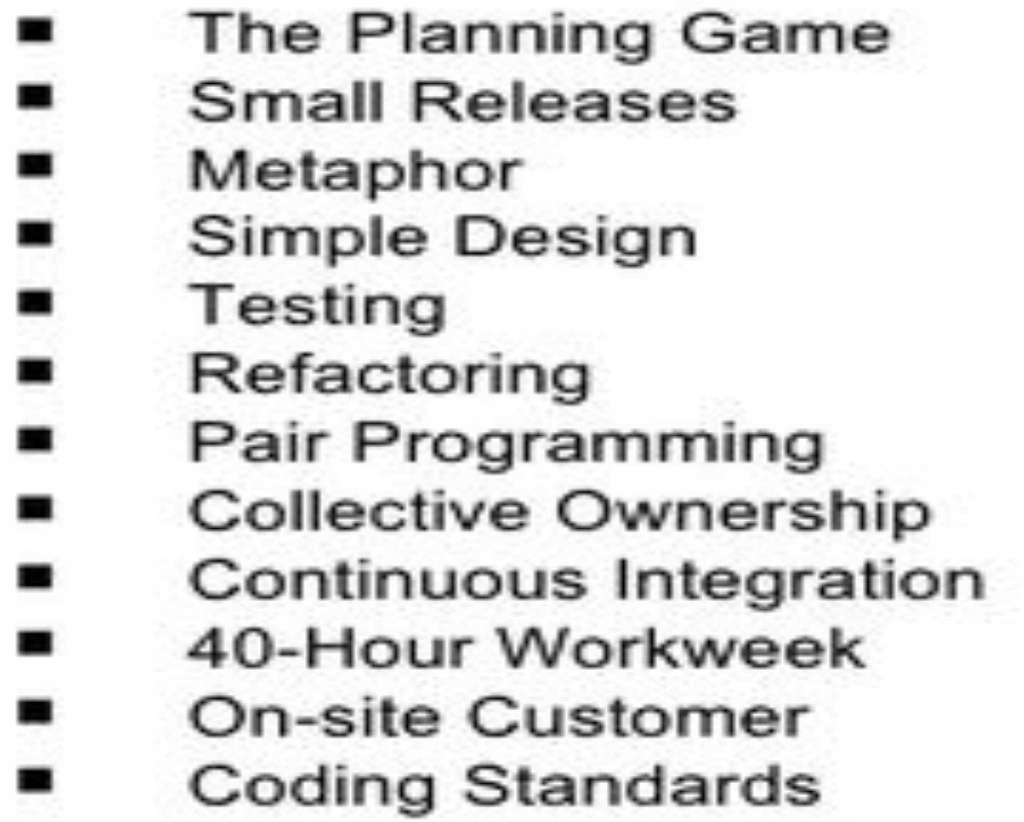
Four Core Values of XP



- Communication
- Simplicity
- Feedback
- Courage

REF : https://www.slideshare.net/bilalhashmishah/extreme-programming-12836919

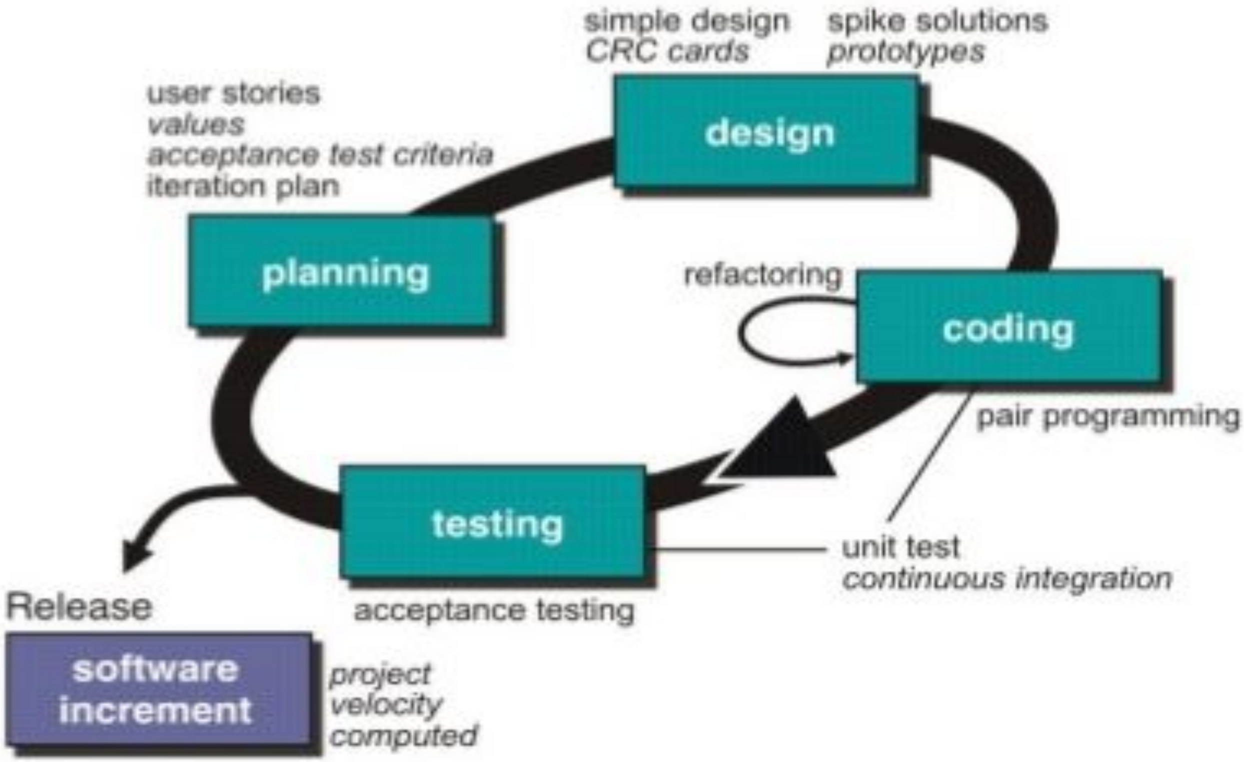
XP Practices



- The Planning Game
- Small Releases
- Metaphor
- Simple Design
- Testing
- Refactoring
- Pair Programming
- Collective Ownership
- Continuous Integration
- 40-Hour Workweek
- On-site Customer
- Coding Standards

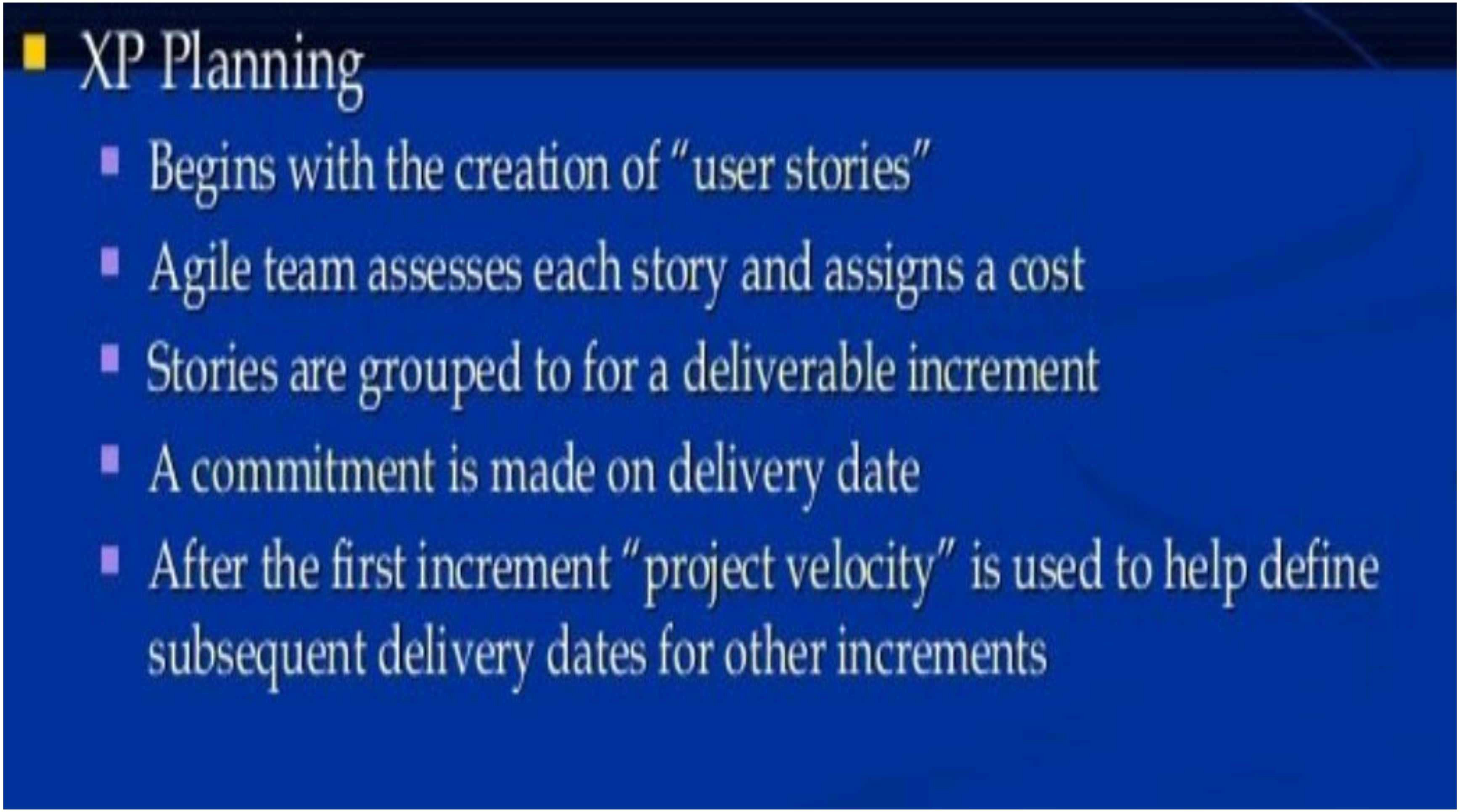
REF : https://www.slideshare.net/bilalhashmishah/extreme-programming-12836919

Extreme Programming Process



Ref: http://www.thedynamicdomain.com/extreme-programming.aspx

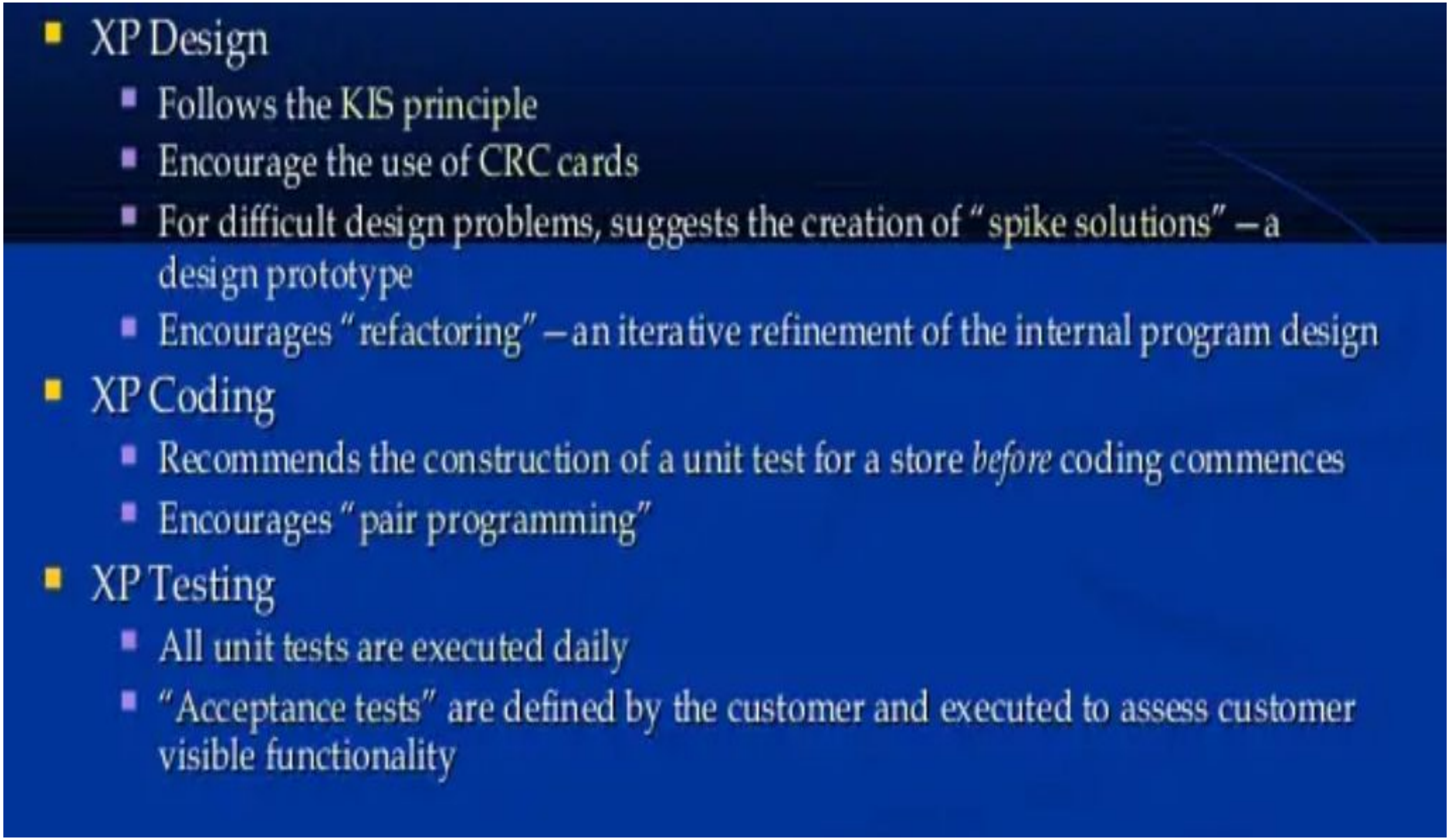
Extreme Programming Process



- XP Planning
  - Begins with the creation of "user stories"
  - Agile team assesses each story and assigns a cost
  - Stories are grouped to for a deliverable increment
  - A commitment is made on delivery date
  - After the first increment "project velocity" is used to help define subsequent delivery dates for other increments

Ref : https://www.slideshare.net/BAMEMON/ch04-agile-development-models

Extreme Programming Process



- XP Design
  - Follows the KIS principle
  - Encourage the use of CRC cards
  - For difficult design problems, suggests the creation of "spike solutions" – a design prototype
  - Encourages "refactoring" – an iterative refinement of the internal program design
- XP Coding
  - Recommends the construction of a unit test for a store *before* coding commences
  - Encourages "pair programming"
- XP Testing
  - All unit tests are executed daily
  - "Acceptance tests" are defined by the customer and executed to assess customer visible functionality

Ref : https://www.slideshare.net/BAMEMON/ch04-agile-development-models

References

- https://www.unf.edu › ~broggio › cen6016 › classnotes › 6a.Agile Software.
- www.uml.org.cn › softwareprocess
- https://www.unf.edu › ~broggio › cen6016 › classnotes › Lecture 12 - Agi...

References

- [1]. Abrahamsson P, Salo O and Ronkainen J. Agile software development methods (Review and analysis).
- [2]. Scott W Ambler. Agile model driven development.
- [3]. Cohen D, Lindvall M, Costa P. Agile software development.
- [4]. http://en.wikipedia.org/wiki/Agile_Modeling.
- [5]. http://en.wikipedia.org/wiki/Extreme_Programming.
- [6]. http://en.wikipedia.org/wiki/Agile_Unified_process.
- [7]. http://en.wikipedia.org/wiki/Scrum_28development29.

CS435: Introduction to Software Engineering

Dr. M. Zhu

Chapter 3

■■ Agile Development

Slide Set to accompany

Software Engineering: A Practitioner's Approach, 7/e

by Roger S. Pressman

Thank You

More Terminology Used in Scrum:

- **Tasks**: Added to story at beginning of a sprint and broken down into hours.
  - Each task should not exceed 12 hours, but it's common for teams to insist that a task take no more than a day to finish.

- **Definition of Done** (DoD): The exit-criteria used to determine whether a product backlog item is complete.

- In many cases the DoD requires that all regression tests should be successful.

More Terminology Used in Scrum:

- **Velocity**: The total effort a team is capable of in a sprint. The number is derived by adding all the **story points** from the last sprint's stories/features.

- This is a **guideline** for the team and assists them in understanding **how many stories** they can do in a sprint.

- **Impediment**: Anything that prevents a team member from performing work as efficiently as possible.

Conclusions

- The professional goal of every software engineer, and every development team, is to deliver the highest possible value to our employers and customers.
  - And yet, our projects fail, or fail to deliver value, at a dismaying rate.
- Though well intentioned, the **upward spiral of process inflation** is culpable for at least some of this failure.
- The principles and values of agile software development were formed as a way
  - to help teams break the cycle of process inflation, and
  - to focus on simple techniques for reaching their goals.

- At the time of this writing there were many agile processes to choose from. These include
  - SCRUM,
  - Crystal,
  - Feature Driven Development (FDD),
  - Adaptive Software Development (ADP), and most significantly,
  - Extreme Programming (XP).
  - Others…

Information Radiator

- "Two characteristics are key to a good information radiator. The first is that the information changes over time. This makes it worth a person's while to look at the display... The other characteristic is that it takes very little energy to view the display."