Agile Development

Common Fears for Developers

The project will produce the wrong product.

The project will produce a product of inferior quality.

The project will be late.

We’ll have to work 80-hour weeks.

We’ll have to break commitments.

We won’t be having fun.

The Manifesto for Agile Software Development

“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

That is, while there is value in the items on the right, we value the items on the left more.”

-- Kent Beck et al.

What is “Agility” ?

Effective (rapid and adaptive) response to change

Effective communication among all stakeholders

Drawing the customer onto the team

Organizing a team so that it is in control of the work performed

Yielding …

Rapid, incremental delivery of software

An Agile Process

Driven by customer descriptions of what is required (scenarios)

Recognizes that plans are short-lived

Develops software iteratively with a heavy emphasis on construction activities

Delivers multiple ‘software increments’

Adapts as changes occur

Principles of Agility

Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

Welcome changing requirements, even late in development. Agile processes harness change for the customer’s competitive advantage.

Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter time scale.

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

Principles of Agility

Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

Working software is the primary measure of progress.

Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

Principles of Agility

Continuous attention to technical excellence and good design enhances agility.

Simplicity - the art of maximizing the amount of work not done - is essential.

The best architectures, requirements, and designs emerge from self-organizing teams.

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

Extreme Programming (XP)

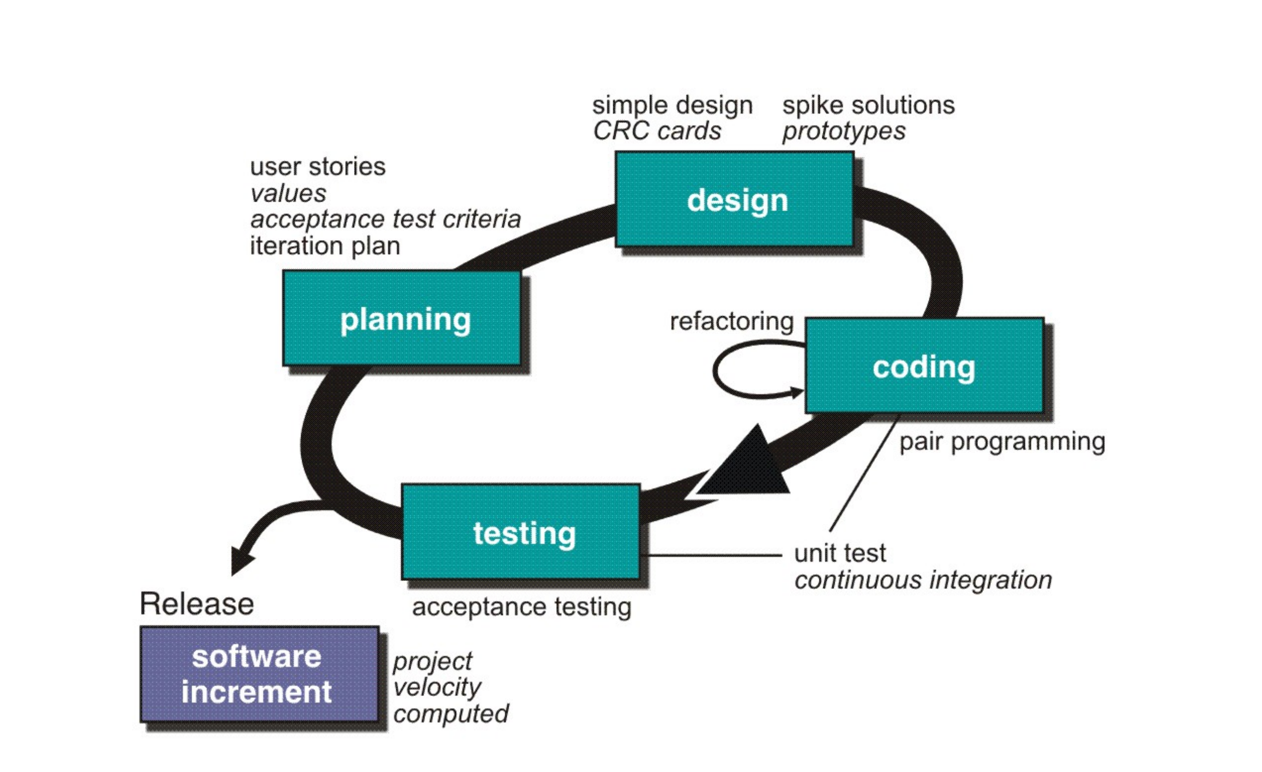
The most widely used agile process, originally proposed by Kent Beck [BEC99]

XP uses an object-oriented approach as its preferred development paradigm

Defines four (4) framework activities

- Planning
- Design
- Coding
- Testing

Extreme Programming (XP)



XP - Planning

Begins with the creation of a set of stories (also called *user stories*)

Each story is written by the customer and is placed on an index card

The customer assigns a value (i.e. a priority) to the story

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

XP - Design

Follows the KIS (keep it simple) principle

Encourage the use of CRC (class-responsibility-collaborator) cards

For difficult design problems, suggests the creation of "*spike solutions*"—a design prototype

Encourages "*refactoring*"—an iterative refinement of the internal program design

Design occurs both before and after coding commences

XP - Coding

Recommends the construction of a series of **unit tests** for each of the stories before coding commences

Encourages "*pair programming*"

- Mechanism for real-time problem solving and real-time quality assurance
- Keeps the developers focused on the problem at hand

Needs continuous integration with other portions (stories) of the s/w, which provides a "**smoke testing**" environment

XP - Testing

Unit tests should be implemented using a framework to make testing automated. This encourages a regression testing strategy.

Integration and validation testing can occur on a daily basis

Acceptance tests, also called *customer tests*, are specified by the customer and executed to assess customer visible functionality

Acceptance tests are derived from user stories

Scrum

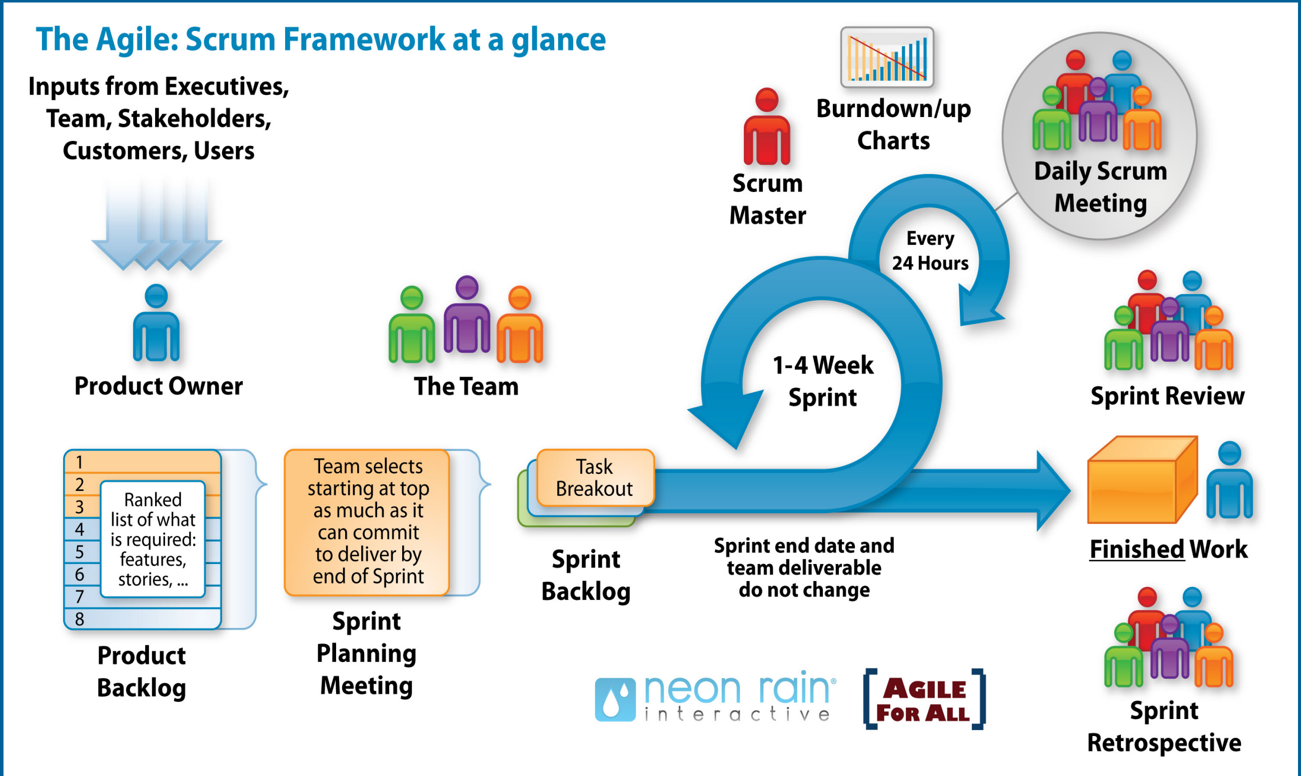
Scrum

A software development method Originally proposed by Schwaber and Beedle (an activity occurs during a rugby match) in early 1990.

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.

Scrum



The Agile: Scrum Framework at a glance

Inputs from Executives, Team, Stakeholders, Customers, Users

Product Owner

The Team

Scrum Master

Burndown/up Charts

Daily Scrum Meeting

Every 24 Hours

1-4 Week Sprint

Sprint Review

1 2 3 4 5 6 7 8

Ranked list of what is required: features, stories, ...

Product Backlog

Team selects starting at top as much as it can commit to deliver by end of Sprint

Sprint Planning Meeting

Task Breakout

Sprint Backlog

Sprint end date and team deliverable do not change

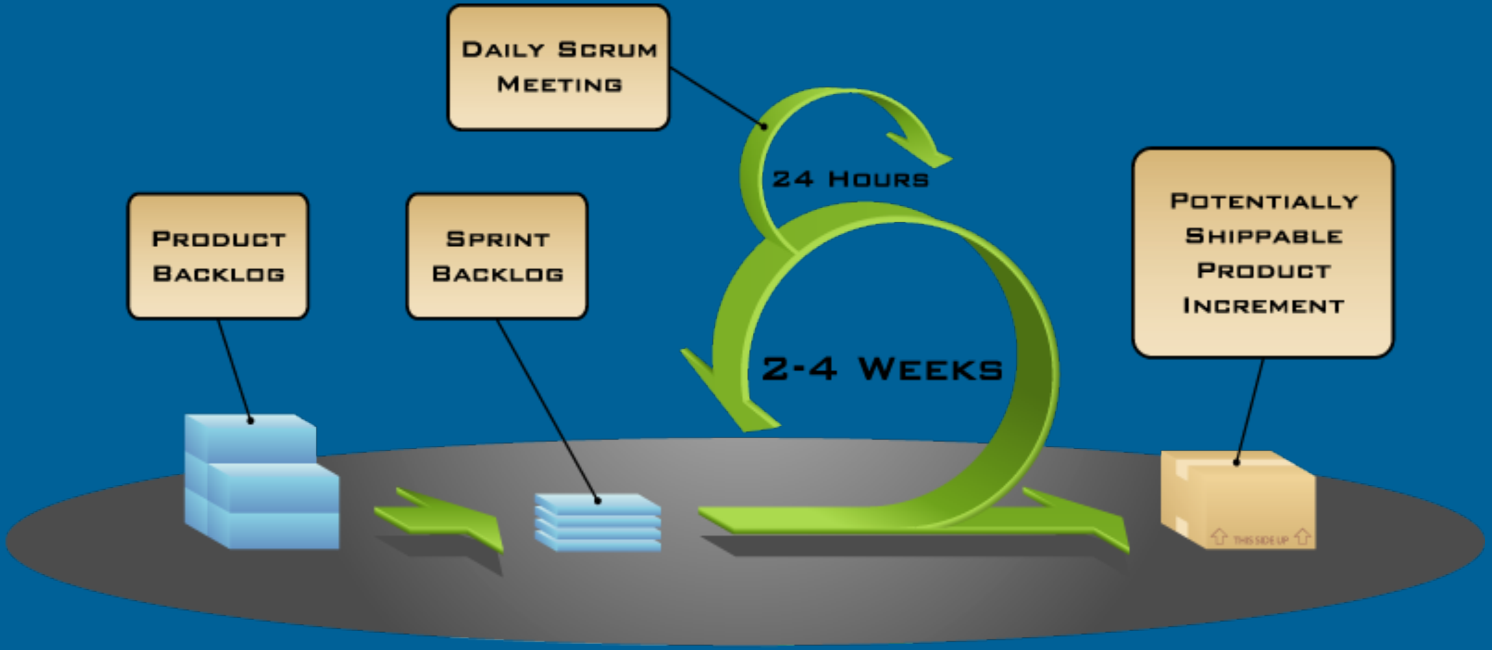
Finished Work

Sprint Retrospective

neon rain interactive

[AGILE FOR ALL]

How Scrum Works?



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

Scrum's core values

Commitment. Because we have great control over our own destiny, we become more committed to success.

Focus. Because we focus on only a few things at a time, we work well together and produce excellent work. We deliver valuable items sooner.

Openness. As we work together, we practice expressing how we're doing and what's in our way. We learn that it is good to express concerns so that they can be addressed.

Respect. As we work together, sharing successes and failures, we come to respect each other and to help each other become worthy of respect.

Courage. Because we are not alone, we feel supported and have more resources at our disposal. This gives us the courage to undertake greater challenges

Other Agile Processes

Adaptive Software Development (ASD)

Dynamic Systems Development Method (DSDM)

Crystal

Feature Driven Development

Agile Modeling (AM)