**Vocabulary**

**Model-View-Controller (MVC)**

Model–view–controller (usually known as MVC) is a software design pattern commonly used for developing user interfaces which divides the related program logic into three interconnected elements. This is done to separate internal representations of information from the ways information is presented to and accepted from the user. Following the MVC architectural pattern decouples these major components allowing for code reuse and parallel development.

**Networking (NET)**

A computer network is a digital telecommunications network for sharing resources between nodes, which are computing devices that use a common telecommunications technology. Data transmission between nodes is supported over data links consisting of physical cable media, such as twisted pair or fiber-optic cables, or by wireless methods, such as Wi-Fi, microwave transmission, or free-space optical communication.

**Object-Oriented Programming (OOP)**

A programming paradigm based on the concept of "objects", which can contain data, in the form of fields (often known as attributes or properties), and code, in the form of procedures (often known as methods).

**Progressive Web Application (PWA)**

A progressive web application (PWA) is a type of application software delivered through the web, built using common web technologies including HTML, CSS and JavaScript. It is intended to work on any platform that uses a standards-compliant browser. Functionality includes working offline, push notifications, and device hardware access, enabling creating user experiences similar to native applications on desktop and mobile devices.

**Project Management (PM)**

The practice of initiating, planning, executing, controlling, and closing the work of a team to achieve specific goals and meet specific success criteria at the specified time. The primary challenge of project management is to achieve all of the project goals within the given constraints.

**Quality Assurance (QA)**

A means of monitoring the software engineering processes and methods used to ensure proper quality. This is accomplished by many and varied approaches. It may include ensuring conformance to one or more standards.

**Representational State Transfer (REST and RESTful APIs)**

A software architectural style that defines a set of constraints to be used for creating Web services. Web services that conform to the REST architectural style, called RESTful Web services, provide interoperability between computer systems on the Internet.

**Responsive Web Design (RWD)**

An approach to web design that makes web pages render well on a variety of devices and window or screen sizes.

**Service-Oriented Architecture (SOA)**

A style of software design where services are provided to the other components by application components, through a communication protocol over a network.

**Single-page Application (SPA)**

A web application or web site that interacts with the user by dynamically rewriting the current page rather than loading entire new pages from a server. This approach avoids interruption of the user experience between successive pages, making the application behave more like a desktop application.

**Software Development (DevOps)**

A process for planning, creating, testing, and deploying an information system. The systems development life cycle concept applies to a range of hardware and software configurations, as a system can be composed of hardware only, software only, or a combination of both.[2] There are usually six stages in this cycle: requirement analysis, design, development and testing, implementation, documentation, and evaluation.

**User Interface and Experience (UI and UX)**

User interface (UI) design is the process of making interfaces in software or computerized devices with a focus on looks or style. Designers aim to create designs users will find easy to use and pleasurable. UI design typically refers to graphical user interfaces but also includes others, such as voice-controlled ones. User experience (UX) refers to any interaction a user has with a product or service. UX design considers each and every element that shapes this experience, how it makes the user feel, and how easy it is for the user to accomplish their desired tasks.

**Open Web Application Security Project (OWASP)**

An online community that produces freely-available articles, methodologies, documentation, tools, and technologies in the field of web application security.

**Payment Card Industry Data Security Standard (PCI DSS)**

An information security standard for organizations that handle branded credit cards from the major card schemes.

**Health Insurance Portability and Accountability (HIPAA)**

Created primarily to modernize the flow of healthcare information, stipulate how Personally Identifiable Information maintained by the healthcare and healthcare insurance industries should be protected from fraud and theft, and address limitations on healthcare insurance coverage.

**Protected Health Information (PHI)**

Under the US law is any information about health status, provision of health care, or payment for health care that is created or collected by a Covered Entity (or a Business Associate of a Covered Entity), and can be linked to a specific individual. This is interpreted rather broadly and includes any part of a patient's medical record or payment history.

**Program Management Institute (PMI)**

A global nonprofit professional organization considered as the world’s leading association for those who consider project, program or portfolio management their profession.

**Ad Hoc**

**Agile**

Agile software development comprises various approaches to software development under which requirements and solutions evolve through the collaborative effort of self-organizing and cross-functional teams and their customer/end user.

* Acceptance Criteria: The conditions under which a piece of work may be held to be complete and fit for potential release.
* Acceptance Test Driven Development (ATDD): A development approach in which acceptance criteria are validated by testing. The necessary tests are implemented first, and development work must then be done which is sufficient to satisfy them.
* Actionable Metrics: Empirical measures which are taken in order to assess value, and which may be used to inspect and adapt a product and/or a development process so that better value can be delivered. Compare Vanity Metrics.
* Adaptation: The modification of a product, role, event, or artefact in light of inspection. In an agile way of working, adaptation ought to occur as close as possible to the time and place of inspection in order to minimize waste.
* Agile Coach: A suitably skilled person, such as a Scrum Master, who can teach, train, mentor, or otherwise communicate agile change and its associated ways-of-working to others. Compare Agile Sponsor.
* Agile Sponsor: An authoritative person within an organization who communicates the necessity for deep and pervasive change, and who thus ensures that an agile transformation and its associated coaching initiatives are able to overcome organizational gravity.
* Agile Transformation: The process of realigning an organization towards the timely and empirical delivery of emergent value, in order that innovative potential might be harnessed. The change required is typically deep and pervasive. See also Agile Coach, Agile Sponsor.
* Anti-pattern: A behavioral or organizational pattern or practice which is contrary to an agile way of working, and which suggests the need for more appropriate patterns or practices.
* Artefact: An input or output of an agile process. In Scrum there are three artefacts: the Sprint Increment, the Sprint Backlog, and the Product Backlog.
* Automation: The elimination of manual work, particularly with regard to the building, integration, testing, and deployment of product increments.
* Automated Build: The compilation and linking of system elements without need for manual initiation or intervention, typically in response to a trigger such as a code check-in or a timed event.
* Automated Testing: The testing of system elements, including functional and integration tests, without need for manual initiation or intervention. The execution of an automated test harness may be triggered by the completion of an automated build. A high degree of test automation is usually needed to make regression testing practicable. See also: Regression Testing
* Avatar: A motif, image, or other symbol which designates a particular team member, and which is attached to a representation of work in progress in order to communicate that the individual is helping to complete it.
* Backlog: An ordered list of items to be worked on by one or more teams, and which thus allows requirements to be satisfied in a timely and effective manner.
* Backlog Refinement: The process of adding detail, order, and estimates to items of inventory in a backlog. Items will have been refined to a sufficient degree once they are ready to be worked on. Refinement is best conducted by those who will eventually do the work.
* Batch: A group of items drawn from a backlog which are processed jointly, rather than individually. Small batch sizes allow work to be processed comparatively quickly and for value to be evidenced earlier. Limiting inventory, such as work-in-progress, to a certain number of items is a common way of restricting batch size.
* Behavior Driven Development (BDD): An approach to development in which the satisfaction of practical usage scenarios and their corresponding acceptance criteria is paramount. See also: Acceptance Test Driven Development.
* Blocked: The state of a work item when it can no longer be progressed due to an impediment which team members are unable to resolve. A Scrum Master will pro-actively seek to remove impediments which block team progress.
* Burn-down Chart: A chart which shows the amount of work which is thought to remain in a backlog. Time is shown on the horizontal axis and work remaining on the vertical axis. As time progresses and items are drawn from the backlog and completed, a plot line showing work remaining may be expected to fall. The amount of work may be assessed in any of several ways such as user story points or task hours. Work remaining in Sprint Backlogs and Product Backlogs may be communicated by means of a burn-down chart. See also: Burn-up chart.
* Burn-up Chart: A chart which shows the amount of work which has been completed. Time is shown on the horizontal axis and work completed on the vertical axis. As time progresses and items are drawn from the backlog and completed, a plot line showing the work done may be expected to rise. The amount of work may be assessed in any of several ways such as user story points or task hours. The amount of work considered to be in-scope may also be plotted as a line; the burn-up can be expected to approach this line as work is completed.
* Business as Usual (BAU): Work which is not temporary in nature, and which is thus not typical of a project. Ongoing product and service support, including maintenance tasks, are often considered to be BAU.
* Cadence: A regular cycle of inspection and adaptation. Artefacts, progress towards a goal, engineering practices, and market conditions may be inspected and adapted on cadence. Scrum teams observe a daily cadence marked by a Daily Scrum in which progress towards a Sprint Goal is inspected, and the Sprint Backlog may be adapted accordingly. The Sprint itself is a longer cadence during which a product increment may be released, and the conditions and engineering practices in which it is delivered may be evaluated. See also: Daily Scrum, Sprint Planning, Sprint Review, Sprint Retrospective.
* Champion: A Development Team member who is competent to represent a parent organization’s stake in engineering practices and standards. Security considerations, data integrity, branding, and test quality may be significant to the wider organization or its stakeholders.
* Coherence: The quality of the relationship between certain work items which may make them worthy of consideration as a whole, such as a Minimum Viable Product, Minimum Usable Subset, or other feature worthy of release. A Sprint Goal should give coherence to those items planned from a Product Backlog into a Sprint Backlog.
* Collaboration: The quality of teamwork which allows a goal to be focused on and achieved by self-organizing team members, by means of their joint effort.
* Commitment: The inviolable and indivisible purpose behind collaborative work. In Scrum, Development Teams will commit to a Sprint Goal, during Sprint Planning, and which will not change throughout the duration of the Sprint time-box.
* Continuous Delivery: The automated elevation of increments of release quality into a production environment. See also: continuous deployment, continuous integration.
* Continuous Deployment: The automated elevation of work done into successive environments which approach production quality. See also: automation, automated build, automated testing. See also: continuous delivery, continuous integration.
* Continuous Integration: The automated integration of work done into an increment of potential release quality. See also continuous delivery, continuous deployment.
* Cross Functional: The quality of a team which allows them to complete work-in-progress by themselves, and without recourse to skills or resources outside the team. Compare with Skills Silo.
* Daily Scrum: A daily inspect-and-adapt event conducted by Scrum Development Teams, in which they re-plan their progress towards their Sprint Goal.
* Defect: A non-conformance of system performance with expected results. A form of waste which can be minimized by test-driven development.
* Definition of Done: Stipulated conditions which must hold for a product increment to be considered finished, and of release quality.
* Definition of Ready: Stipulated conditions which must hold for a work item to be actioned by a team, such as the quality of its acceptance criteria or its degree of scope.
* Dependency: The relationship between artefacts, roles, or items of work which may inhibit their progress unless they are decoupled, or those relationships are taken into consideration.
* Development Team: Those persons who, by collaborative effort and shared commitment, may develop an increment of value and of usable quality. A Scrum Development Team ought to be cross-functional and should number between three and nine members.
* DevOps: The bridging of development work and operational support responsibilities in a cross-functional team. A DevOps capability is typically facilitated through a high level of automation.
* Emergence: The incremental process through which complex qualities are better understood and gradually brought into focus. In an agile way-of-working, an emergent system is understood by the inspection and adaptation of empirical evidence, including that presented by the incremental delivery of value, often on cadence.
* Empiricism: The evaluation of a system in terms of real-world impact and the value it is observed to provide, rather than by an alternative or proxy measure. Compare with Vanity Metrics.
* Empowerment: The removal of restrictions on a team’s ability to collaborate and self-organize, including dependencies and skill-silos, so that they can meet an agreed goal or commitment.
* Engineering Practices: The ways-of-working used by a Development Team to build a product increment in an agile manner.
* Engineering Standards: The best practices which one or more Development Teams will observe in order to create an increment of release quality.
* Epic: A user story or requirement which aggregates multiple other such items in a way which gives them coherence.
* Escalation: The raising of a problem, such as an impediment, to an authority which is in a position to resolve it. Agile teams ought to be self-organizing and sufficiently empowered to render escalation beyond the team unnecessary. A Scrum Development Team may escalate impediments to their Scrum Master for resolution.
* Estimation: The sizing of work, typically on a backlog, by those who are likely to perform that work. In Scrum, estimation allows a Development Team to ascertain how much work it can plan into a Sprint, and for forecasts to be made regarding how much work is thought to remain at any given point.
* Exploratory Testing: The testing of work with the intention of uncovering defects which have not yet been exposed by automated means. Exploratory testing is typically manual and ad-hoc in nature.
* Fail Early: The elimination of future waste by calling a halt to an initiative which has failed to demonstrate empirical value, and before any further waste is allowed to accumulate.
* Fibonacci Sequence: A sequence of numbers commonly used or approximated for estimation purposes, in which each successive number beyond the second number is the sum of the two prior numbers, viz: 1, 2, 3, 5, 8, 13. The use of this sequence reduces the opportunity for false precision when estimating large pieces of work.
* Forecast: A plan or projection based on empirical evidence, such as the rate of work completed.
* Huddle: See Daily Scrum.
* Increment: A usable implementation of an emergent product which is made available at a certain point in time. In Scrum, product increments are held to be cumulative, such that the latest increment will incorporate the value of all prior increments.
* Incremental Development: The provision of an emergent product or service by means of successive development efforts, often on cadence, each of which provides a usable increment of value.
* Information Radiator: A mechanism through which timely and useful information can be pulled on-demand by interested observers, and relied on for its transparency and accuracy. The most common type of information radiator found in agile practice is a Scrum Board or Kanban Board.
* Innovation: The process of discovering or creating new market opportunities and satisfying them. Such opportunities may become apparent through emergence, and their exploitation may require the use of agile practices. Innovation which compromises existing markets is sometimes referred to as being disruptive.
* Inspection: The process of assessing the condition of a product, role, event, or artefact in response to a triggering condition, so that it may be adapted and improved. Inspection may be triggered by the detection of an error or by the completion of work; either condition may represent an opportunity to reflect and improve. In an agile way of working, adaptation ought to occur as close as possible to the time and place of inspection in order to minimize waste.
* Integration: The combining of completed work items from various sources into an increment of demonstrable release quality. At scale, the deliverables of multiple teams may need to be integrated before a product increment can be released. See also: automation.
* Inventory: Work which is being held in a batch, such as a backlog or as work-in-progress, and which is subject to depreciation until the value of that work is released. See also: batch.
* INVEST Criteria: The qualities of a user story or similar item of inventory which make it independent of other items, negotiable as an element of scope, valuable to stakeholders, estimable by those doing the corresponding work, small enough to be completed by them in a timely manner, and testable in order to prove its fitness for release.
* Iteration: A time-box of fixed duration and regular occurrence during which one or more increments of value are developed and made available for immediate release, and at least one opportunity for inspection and adaptation is presented.
* Just-in-Time: The synchronization of demand with supply so that batch sizes are minimized, and waste caused by inventory depreciation is reduced.
* Kaizen: Improvement, usually taken to mean improvement in a gradual sense by means of empiricism, transparency, inspection, and adaptation.
* Kanban: A closed economy of production in which work in progress is represented by a finite number of tokens, and the delivery of value is optimized by inspection, adaptation, and the ongoing reduction of waste.
* Kanban Board: An information radiator which shows items of inventory being processed in Kanban-fashion, typically by representing them by means of cards, and showing their progress through work stations in which value is added. Uneven flow and other forms of waste can thereby be highlighted and acted upon.
* Latency: The time it takes for a work item, following its acceptance by a Development Team, to be delivered as a usable increment of value.
* Lean Startup: An organization developing a product, service, or business model which enables sustainable growth, and which minimizes waste in order to do so before resources expire.
* Leap-of-Faith: The time, money, or other resources which must be invested in a development effort before empirical proof of value is delivered. In Scrum, the leap-of-faith is minimized to one Sprint’s worth of investment.
* Little’s Law: The relationship between latency, throughput, and work-in-progress, such that WIP = latency x throughput
* Management by Exception: The empowering of teams to self-organize and otherwise operate within certain tolerances. If those tolerances are exceeded then management will be notified and only at that point will they become actively involved. A Scrum Master may notify management by exception if he or she is unauthorized or otherwise unable to resolve impediments which affect the team.
* Metrics: The key indicators and measures which allow progress to be assessed, and for inspection and adaptation to occur. The best metrics are based on empirical evidence and allow improving action to be taken.
* Minimum Usable Subset: The fewest work items which must be completed in order to have an increment which, when released, is usable by stakeholders for a coherent purpose.
* Minimum Viable Product: Either a Minimum Usable Subset, or in Lean Startup terms, the smallest increment which must be released in order to frame and test an hypothesis about the value which really ought to be delivered.
* Nexus: A collaborative group of Scrum Teams who work together on the same product, planning and drawing work from the same Product Backlog, and who create an integrated and tested increment of release quality with respect to a commonly observed Sprint cadence.
* Non-Functional Requirements: Those requirements germane to a system’s scope which are held to be invariant, which cannot be prioritized in relation to functional requirements for incremental delivery, and which, being applicable to every increment, may be indicative of the system’s overarching qualities.
* One Point One Card: A very simple estimation technique in which each item of backlog inventory is held to be of equivalent magnitude (in effect, one point). The use of this technique can encourage the refinement of small pieces of inventory which are less likely to be impeded, and which may thus allow value to be made available earlier. However, it may also encourage a technical focus which makes business prioritization of such items difficult. Where this estimation technique is used, throughput and velocity can be expected to resolve to the same measure.
* Operating Model: The model with reference to which an organization delivers value to its stakeholders. In an agile way of working, an agile framework such as Scrum may be used to define an operating model or an essential part of one.
* Organizational Gravity: The tendency of people within an organization to revert to traditional ways-of-working in response to an agile transformation initiative, and to present defensive behaviors when faced with the expectation of such change. The strength of agile sponsorship is a critically important factor in overcoming organizational gravity. See also: agile transformation, agile sponsor.
* Pair Programming: An engineering practice in which two Development Team members will collaborate in the same programming activity, constantly checking and assisting each other’s work, discussing design decisions, and providing mutual encouragement and immediate peer review.
* Pattern: A common behavior or way-of-working which can be abstracted and reused in order to improve agile practice.
* Peer Review: An engineering practice in which a Development Team member will review the work done by another team member, in order to provide a quality check. Delay and depreciation can be reduced if this occurs in a pair programming context.
* Personas: A personification of a system stakeholder, either real or imaginary, by means of which system requirements can be elicited and described.
* Pivot: A significant adaptation to a new value proposition, architecture, technology, or market, in light of inspection and empirical evidence obtained, and which requires validation by a new MVP. Commonly associated with Lean Startup practices, a pivot ought to minimize any leap-of-faith assumptions regarding the adaptation.
* Planning Poker: A relative estimation technique using playing cards enumerated with a Fibonacci-like sequence. Each member of the Development Team will get a set of cards, and will use them to indicate the relative size of the backlog items being considered. See also: product backlog refinement.
* Potentially Releasable: The quality of an increment which asserts that no further work need be done for its use by stakeholders in the manner intended.
* Prime Directive: A pre-condition which must hold for a Sprint Retrospective to be conducted, in which team members acknowledge each other’s competence to be there and agree to respect each other’s opinions without rancour. See also: Scrum values.
* Product Backlog: A backlog placed under the stewardship of a Product Owner, who uses it to frame and order the scope of an emergent product so that maximum value can be delivered by one or more development teams.
* Product Backlog Refinement: See backlog refinement.
* Product Owner: The authoritative voice representing a product, and the person who is responsible for maximizing the value it provides. The Product Owner is accountable to stakeholders for product value, and represents and balances their interests.
* Proxy Product Owner: An authority to whom a Product Owner may choose to delegate certain responsibilities. Proxy Product Ownership has the potential to cloud authority regarding product scope and/or accountability for product value, and is thus sometimes considered to be an anti-pattern.
* Pull: The demand exerted by consumers of value for completed work, and which encourages development teams to release usable increments at a sustainable pace and which maximizes the value delivered.
* Quality: The characteristics of a product increment or service which render it fit for use by stakeholders.
* Quality of Service: The degree of quality which is appropriate for a particular service provided to stakeholders, such as may be defined by a service level agreement.
* Ready: The state of a backlog item when it is capable of being brought into progress and worked on. See also: backlog refinement.
* Refactoring: The re-organizing of code, system components, or corresponding design elements to improve transparency and ease of understanding. Refactoring does not change the functional characteristics of the system.
* Regression Testing: The execution of all of test cases which assert that a product is fit for use, including those test cases which verify the functionality delivered in earlier product increments.
* Release: The delivery of a product or service increment into a production-quality environment, such that it is available for immediate use by stakeholders.
* Release Plan: A forecast which indicates the increments of value which may reasonably be expected in the future. Stakeholders, or a representative authority such as a Product Owner, may make such forecasts based on the empirical evidence of delivery and the work which is currently thought to remain in a Product Backlog.
* Relative Estimation: The sizing of backlog items in terms of their comparative scale. This contrasts to absolute estimation in terms of projected cost or likely time to complete. Relative estimation can be accomplished qualitatively, such as by T-Shirt Sizing (e.g. XS, S, M, L, XL, XXL) or quantitatively, such as by story points. See also: Planning Poker, Fibonacci sequence.
* Requirements: The characteristics which a product or service is expected to demonstrate, including its functional scope, its non-functional properties, and its degree of quality. Where the complexity of a product or service domain is high, requirements may be emergent.
* Scope: The functional and non-functional requirements of a product or service which are expected to be satisfied.
* Scrum: A huddle of team members who have assembled temporarily in order to collaborate and solve a problem, or to otherwise reach a joint agreement (e.g. a Daily Scrum). The Scrum Framework is an agile development approach based on this technique, and which is used for building complex products.
* Scrum Board: An information radiator which shows the progress of a team’s work as it is drawn, actioned, and completed from their Sprint Backlog. A Scrum board may show user stories and/or planned tasks, or both, or some other meaningful representation of the team’s work.
* Scrum Master: The servant-leader for a Scrum Team, who removes impediments to their work, facilitates their progress towards the Sprint Goal, and coaches them and the wider organization in the best use of the Scrum Framework.
* Scrum of Scrums: A Scrum held between multiple Scrum Teams or their representatives, often for the purpose of ensuring that mutual dependencies are resolved and that product integration is not impeded.
* Scrum Team: A self-organizing team of professionals, consisting of a Scrum Master, Product Owner, and a Development Team, who deliver increments of value every Sprint and who inspect and adapt their progress.
* Scrum Values: The characteristics which Scrum Team members seek to internalize and demonstrate as a cultural norm, including Commitment, Focus, Respect, Openness, and Courage. See also: prime directive.
* Self-Organization: The quality of a team which allows it to inspect and adapt its behaviors, without external guidance, so that a goal or purpose can be met. A self-organizing team must be sufficiently cross-functional in terms of the skills it incorporates for planned work to be done. Team members may exhibit a degree of cross-skilling so that each is capable of doing more than one kind of activity.
* Skills Silo: A constraint upon team self-organization which restricts individual team members to performing only one type of activity.
* Sponsorship: The momentum provided by stakeholders for implementing change across an organization. Agile transformation typically requires deep and pervasive change, and a correspondingly strong sponsorship which is communicated across multiple levels. See also: organizational gravity.
* Sprint: A time-box of no more than one calendar month during which a Scrum Team will frame a Sprint Goal, and satisfy that goal by delivering a product increment of release quality. Sprints occur on a regular cadence and permit the inspection and adaptation of progress.
* Sprint Backlog: A forecast of the work a Development Team plans to do in order to meet an agreed Sprint Goal. The backlog may be revised during the Sprint in order to better meet the Goal.
* Sprint Goal: The purpose for conducting a Sprint, framed and agreed during Sprint Planning between the Development Team and the Product Owner. A Sprint Goal will give coherence to the Product Backlog Items selected and planned into the Sprint Backlog. In order for the Sprint Goal to be met, a corresponding increment of value must be developed and made available for release.
* Sprint Planning: The act of negotiating a coherent selection of Product Backlog Items into an achievable Sprint Backlog of work, and the framing of a corresponding valuable Sprint Goal. A Sprint Plan must be agreed by the Development Team and the Product Owner.
* Sprint Retrospective: The inspection and adaptation of a Scrum Team’s way-of-working, at the end of a Sprint, so that future Sprints might be improved. The whole Scrum Team is expected to participate. See also: prime directive.
* Sprint Review: The inspection and adaptation of a Product Increment and the Product Backlog, at the end of a Sprint, so that the characteristics of the product can be evaluated and better understood.
* Sprint Zero: A common anti-pattern in which a false Sprint is conducted for project initialization purposes, and without the delivery of an increment of value. A Sprint must always deliver an increment of value, no matter how small.
* Stakeholder: A party with an interest in a Development Team’s output. In the Scrum Framework, product stakeholder interests are represented by a Product Owner.
* Stand-up: See Daily Scrum.
* Station: A discrete point during the flow of work-in-progress at which value is added. Stations may include test authoring, development, review, and integration. Clear stations can improve transparency over work-in-progress and expose irregularities in flow and other instances of waste. An efficient, self-organizing team will reduce skill-silos so that members can move freely over stations and smooth the progress of work.
* Stewardship: The demonstration by a team member that he or she is a responsible custodian. A Product Owner should demonstrate good stewardship of the Product Backlog, while a Scrum Master should demonstrate good stewardship of the Development Team and its implementation of the Scrum Framework.
* Stop-the-line: An event which triggers immediate inspection-and-adaptation once a problem is detected, and which stops the problem from compounding. Work-in progress is suspended until remedy; the team will focus on providing that remedy and on eliminating the chance of the problem recurring. The ability of self-organizing team members to halt work and correct errors in this manner is instrumental in limiting waste.
* Story Mapping: The correlation of user stories to epics, features, and/or time-boxed opportunities for delivery, such as Sprints. Story mapping techniques can be used to frame Minimum Viable Products or Minimum Usable Subsets, or to propose tentative Sprint Goals. See also: backlog refinement.
* Story Points: A measure used to estimate the relative size of backlog items, such as user stories. See also: relative estimation, Planning Poker, Fibonacci sequence.
* Studio: A creative environment within which the agile delivery of value is fully supported. Stakeholders who wish to use the studio must agree to its terms of engagement.
* Sustainable Pace: The quality of a team to work in a predictable manner at a regular cadence, such that useful stakeholder forecasts can be made regarding progress through a backlog of work.
* Task Board: An information radiator showing the breakdown of work into technical tasks, and the progress of that work over various stations towards completion. See also: Scrum board.
* TDD: Test Driven Development. The authoring of a test prior to the development of a capability which satisfies the condition. The test case can be assumed to fail when first executed. If it passes then it suggests either an error in the test, or the prior existence of a satisfactory capability. Enough development work is then done in order to ensure that the test passes. The code may then be refactored, and the test relied on to make sure that functionality has not been compromised. Also known as Red-Green-Refactor. See also: refactoring.
* Team: A collaborative group of people with a joint sense of purpose, and who are able to demonstrate teamwork. As well as a Scrum Master and a Product Owner, a Scrum Team will further contain a Development Team.
* Technical Debt: The longer-term consequence of poor design decisions. Technical debt is often incurred for the sake of expediency. A robust Definition of Done is instrumental in keeping technical debt under control.
* The Three C’s: Card, Conversation and Confirmation. The three essential properties of a user story, which may be written on a “card". A user story is not a requirement, but rather a placeholder for one or more “conversations” which are held in order to satisfy the requirement. The implementation of the requirement must then be “confirmed”, such as by means of validated acceptance criteria or the successful execution of corresponding tests.
* Three Questions: The planning considerations which each member of a Scrum Development Team must elicit during a Daily Scrum: namely, what they did yesterday to help the team meet the Sprint Goal, what they intend to do today to help meet that goal, and any impediments to their doing so.
* Throughput: The rate at which increments of value are delivered by a Development Team.
* Time-box: A period of fixed maximum duration in which team activities may be carried out. In Scrum there are five time-boxed events: Sprint Planning, the Daily Scrum, the Sprint Review, the Sprint Retrospective, and the Sprint itself which is a time-box containing all other events.
* Tolerances: The remit within which a self-organizing team is empowered to operate. See also: management by exception.
* Transparency: The degree of openness which is afforded to those who have a stake in a process. A transparent process is of critical importance to team members if they are to be able to inspect and adapt it. Transparency, along with inspection and adaptation, is fundamental to an agile way-of-working.
* T-Shirt Sizing: A relative estimation technique which uses qualitative measures rather than quantitative ones to assert the projected size of backlog items. Typical sizes include XS, S, M, L, XL, and XXL. The use of T-Shirt sizing may make the elicitation of metrics, such as a burn-down chart, difficult unless they are mapped to a numerical scheme such as story points. See also: relative estimation.
* Unit Testing: The authoring and execution of test cases which verify the functional correctness of specific units of code, such as functions or methods and their modules or classes. See also: TDD.
* Usability Testing: The verification of system functionality by end-users or their representatives, thereby increasing the degree of assurance that an increment will prove to be of usable quality should it be released.
* User Story: A placeholder for one or more conversations about a possible requirement. User stories are commonly used to represent the functional characteristics of a system from the perspective of an end-user, and will assert one or more acceptance criteria. A Product Backlog may consist largely or entirely of user stories. See also: The Three C’s, acceptance criteria, INVEST.
* Value: The quality of a product or service which makes an investment in it worthwhile. Stakeholders in products or services can have different perspectives regarding the value which ought to be provided. A Product Owner will represent and arbitrate those interests, and seek to maximize the value delivered. A Product Owner has authority over what value means for a product.
* Value Stream: The flow of work-in-progress across discrete stations where value is added. A product or service may incorporate one or more value streams. Development Teams may restrict their work to one value stream at a time in order to achieve focus and reduce waste.
* Vanity Metrics: Measurements which are taken to show progress or value in the best possible light, and which thus fail to provide adequate transparency for the purposes of inspection and adaptation. Compare with actionable metrics.
* Velocity: The rate at which work, expressed in terms of work items which have been relatively sized, is completed by a Development Team. In Scrum, a common measure of velocity is the number of story points which are, on average, reduced from the Product Backlog each Sprint.
* Waste: The avoidable loss of value in a development and delivery process. Seven wastes are frequently attested for: transport costs (i.e. the handover of work), the cost of maintaining concurrently held inventory, the cost of moving work-in-progress around unnecessarily, time spent waiting and being unproductive, overproduction, the over-engineering of work, and depreciation.
* Work in Progress: The work which has been accepted by a team and is pending completion. The limitation of WIP is generally desirable in order to improve throughput and to limit depreciation. See also: Little’s Law.

**Binary tree**

In computer science, a binary tree is a tree data structure in which each node has at most two children, which are referred to as the left child and the right child.

**Caching**

**Canonical**

Confirming to well-established patterns or rules

**Deserialization**

The reverse of that process, taking data structured from some format, and rebuilding it into an object.

**Dogmatic**

Adhering only to principles which are true to hypothesis rather than truths based on experiment or evidence.

**Empirical**

Based on, concerned with, or verifiable by observation or experience rather than theory or pure logic.

**Ingress vs. egress**

Traffic towards you vs. traffic directed towards an external network and originated from inside the host network.

**International Standards Organization (ISO)**

A standards body.

**IPv4 and IPv6**

Internet Protocol version 4 (IPv4) is the fourth version of the Internet Protocol (IP). It is one of the core protocols of standards-based internet working methods in the Internet and other packet-switched networks.

Internet Protocol version 6 (IPv6) is the most recent version of the Internet Protocol (IP), the communications protocol that provides an identification and location system for computers on networks and routes traffic across the Internet.

* Difference between IPv4 & IPv6: Both IP addresses that are binary numbers. IPv4 is 32 bit binary number while IPv6 is 128 bit binary number address. IPv4 address are separated by periods while IPv6 address are separated by colons. Both are used to identify machines connected to a network.

**Paradigm**

A typical example or pattern of something; a model.

**Pragmatic**

Dealing with things sensibly and realistically in a way that is based on practical rather than theoretical considerations.

**Proof of Concept (POC)**

**Read-eval-print loop**

**Route**

**Semantics**

The [linguistic](https://en.wikipedia.org/wiki/Linguistics) and [philosophical](https://en.wikipedia.org/wiki/Philosophical) study of [meaning](https://en.wikipedia.org/wiki/Meaning_(linguistics)) in [language](https://en.wikipedia.org/wiki/Language), programming languages, formal logics, and [semiotics](https://en.wikipedia.org/wiki/Semiotics). It is concerned with the relationship between signifiers—like [words](https://en.wikipedia.org/wiki/Word), [phrases](https://en.wikipedia.org/wiki/Phrase), [signs](https://en.wikipedia.org/wiki/Sign), and [symbols](https://en.wikipedia.org/wiki/Symbol)—and what they stand for in reality, their [denotation](https://en.wikipedia.org/wiki/Denotation).

**Serialization**

The process of converting a data structure or object into a sequence of bits so that it can be stored in a file or memory buffer, or transmitted across a network connection link to be reconstructed later in the same or another computer environment.

**Space complexity**

**Subnet**

In [computer science](https://en.wikipedia.org/wiki/Computer_science), the space complexity of an [algorithm](https://en.wikipedia.org/wiki/Algorithm) or a [computer program](https://en.wikipedia.org/wiki/Computer_program) is the amount of memory space required to solve an instance of the [computational problem](https://en.wikipedia.org/wiki/Computational_problem) as a function of the [size of the input](https://en.wikipedia.org/wiki/Problem_size). It is the memory required by an algorithm to execute a program and produce output.[[1]](https://en.wikipedia.org/wiki/Space_complexity#cite_note-1)

<https://en.wikipedia.org/wiki/Space_complexity>

**Time complexity**

In [computer science](https://en.wikipedia.org/wiki/Computer_science), the time complexity is the [computational complexity](https://en.wikipedia.org/wiki/Computational_complexity) that describes the amount of time it takes to run an [algorithm](https://en.wikipedia.org/wiki/Algorithm). Time complexity is commonly estimated by counting the number of elementary operations performed by the algorithm, supposing that each elementary operation takes a fixed amount of time to perform. Thus, the amount of time taken and the number of elementary operations performed by the algorithm are taken to differ by at most a [constant factor](https://en.wikipedia.org/wiki/Constant_factor).

<https://en.wikipedia.org/wiki/Time_complexity>

**Technical debt**

**Virtual Private Cloud**