UNIVERSITY OF ALBERTA
DEPARTMENT OF COMPUTING SCIENCE

SOFTWARE PROCESSES AND AGILE PRACTICES

GLOSSARY

Glossary

Word	Definition
Acceptance Test	A test that verifies that a product requirement has been satisfied. It can be automated or a script for a human to conduct.
Activity	A set of related tasks.
Ad Hoc Development	Developing software reactively, without a plan.
Adaptation	Responding to changes.
Agile	A philosophy for developing software that is based on the values and principles of the Manifesto for Agile Software Development.
Agile Manifesto	A declaration of the values and principles for software development that follows the Agile philosophy.
Agile Practices	Techniques, rules, or guidelines that are based on the Agile Manifesto to make software development or management more effective.
Agile Unified Process	A methodology that combines the Unified Process model and Agile practices.
Analysis, Requirements	An activity to examine the requirements of a product to ensure, for example, that they are clear, complete, and consistent.
Analyze	Examine methodically and in detail to discover insights or potential improvements.
Architecture	The structure of a software system in terms of components, connections, and constraints.
Backlog	In Scrum, a prioritized list of features and functionalities for a product, maintained by the product owner.
Change-Friendly	A product that is easily adaptable to change.
Client Satisfaction	That the client is happy with the product provided, so it solves their problem, addresses their needs, and meets their expectations.
Coding Standard	A set of guidelines or conventions to be followed by software developers on the formatting, style, and usage for source code in a particular programming language.

Collective Ownership	A practice to encourage anyone on the development team to contribute to any part of the product. The team, not individuals, own the product's successes and failures.
Construction	The phase in the Unified Process model in which the focus is on implementing the product.
Continuous Delivery	A practice to produce valuable software in short iterations to ensure that the software can be reliably released at any time.
Continuous Integration	A practice where developers combine their code frequently into a shared repository to detect build issues or incompatibilities early.
Cross-Functional	The development team is self-contained, consisting of all the skills needed to complete the product.
Cycle	A complete loop of the Unified Process model through its phases.
Design and Implementation Phase	The phase of a software process where you design the organization of the product and implement that design.
Documentation	Documents that describe the product for the developers or end users.
Elaboration	The phase in the Unified Process model in which the focus is on an architectural model or prototype to later drive product implementation.
Elicitation, Requirements	An activity to discover requirements, by interacting with users, clients, and other stakeholders, investigating their needs, and exploring the ideas and features of a potential product.
Estimation	A rough calculation of something, for example, the anticipated effort needed for a requirement or the time needed for a task.
Evolutionary Prototype	A working prototype from a sequence of prototypes, where the first begins with all features in basic form and subsequent prototypes further refine those features.
Exploratory Prototype	A working prototype to learn about the feasibility of a product idea.
Expression, Requirements	An activity to describe the needed product features from the client in a structured way.
External Documentation	Documentation that is intended to be used by users of the product.

Extreme Programming	An Agile methodology that focuses on practices for effective software development, including pair programming and test-driven development.
Illustrative Prototype	A simple prototype to depict product ideas in a low-fidelity form, potentially on disposable media.
Inception	The initial phase in the Unified Process model in which key requirements are outlined and the business case, scope, and risks are identified.
Incremental Prototype	A working prototype from a sequence of prototypes, where the first begins with a core set of features and subsequent prototypes add more features as resources permit. Features are triaged into categories of “must do”, “should do”, and “could do” to determine the order of release.
Input Work Product	A work product that is required for an activity to occur.
Inspection	A procedure to review a work product for flaws.
Integration	Combining software written by different developers into a coherent whole.
Interface	A boundary across which two separate entities communicate, for example, the connection between two software components or the visual presentation that an end user interacts with to use the product.
Internal Documentation	Documentation that is intended to be used by members of the development team for a product.
Invariant	A property that remains unchanging.
Iteration	One instance of a repetition.
Iterative Process	A kind of process with repeated sequences of phases.
Kanban	In software development, a visual way to organize and track required work through their stages of completion.
Life Cycle	The life span of a product.
Life Cycle Process	A process to develop and manage a software product from its initial conception to its retirement.
Linear (Process) Model	A conceptual representation of a linear process.

Linear Process	A kind of process that advances from one phase to another sequentially and only when the previous phase has completed.
Managed	That processes and practices are followed to organize the work of everyone involved in the software project.
Management, Requirements	An activity whereby requirements are organized for reference and reuse, for example, ensuring requirements are traceable to tests and source code.
Methodology	A defined set of practices to perform software development.
Metric	A way to measure that a product or process meets some quality or characteristic.
Metric Data	Data acquired from calculating metrics on a product or process.
Microsoft Daily Build	A practice from Microsoft in which all software must be integrated at the end of the day.
Monitoring	The tracking of a project’s progress and the product’s quality.
Output Work Product	A work product that is generated by an activity.
Pair Programming	A practice where two developers work side-by-side on a task at the same computer.
Parallel Process	A kind of process in which activities can happen across phases and concurrently with other activities.
Phase	A distinct period or stage in a process.
Planning	A procedure done to make plans that organize and schedule upcoming work.
Practices	Techniques, rules, or guidelines to make software development or management more effective.
Prioritization, Requirements	An activity to organize the list of requirements based upon what is of higher value and should be completed earlier.
Process	An organization of the development of software into distinct phases or stages.
Process Model	A conceptual representation of the phases of a process.

Product Owner	In Scrum, the role responsible for the product backlog.
Project	An endeavour to complete a product including its planning, development, operation, and evolution.
Project Management Phase	Consists of parallel and ongoing activities to manage and plan work throughout development.
Prototype	A preliminary form or version of the software product, from which other forms are developed.
Refactoring	A practice to restructure the internal design of the code without changing its behaviour, to allow future changes to be easier.
Requirement	A specific description of a need, such as a desired capability to be implemented in the product.
Resource	Something that improves, advances, or funds a task.
Retrospectives	A way to reflect and review the development of a product and its process to identify lessons learned to improve future work.
Review	To reflect on something and determine what was good and what can be improved upon.
Risk Management	Identifying and mitigating risks before they occur.
Risk Plan	A course of action that outlines the solution if an issue should occur. Also known as an action plan.
Role	A job-related activity that a person takes on.
Sawtooth Model	A linear process model, with specific phases that involve the client.
Scrum	An iterative and incremental Agile methodology for managing software development.
Scrum Events	In Scrum, occurrences facilitated by the scrum master, including sprints, sprint planning, daily scrums, sprint review, and sprint retrospective.
Scrum Master	In Scrum, the role responsible for organizing and facilitating Scrum practices by the product owner and the development team.
Scrum Team	Consists of everyone responsible for making a software product using Scrum practices, including the product owner, the scrum

	master, and development team members.
Self-Organizing	A development team that is empowered to decide how to implement the product and determine on their own who will take on each required task.
Software Engineering Activity	See Activity.
Specification Phase	The phase of a process when the idea for the product is conceived, including a definition of what the product does.
Spiral Model	A risk-driven process model by Boehm that iterates through four phases.
Sprint	In Scrum, an iteration over a short time period, whereby an increment of working software is delivered to the client at the end for feedback.
Standardize	To adopt development practices that ensure consistency or compatibility.
Sub-Process	A smaller process within a larger process.
Sub-Teams	A smaller team within a larger team.
System Metaphor	A practice to explain the product simply to someone else, through familiar concepts or by analogy.
Task	A small, manageable unit of work to be completed.
Task Dependency	A relationship that specifies the ordering of tasks.
Test-Driven Development	A practice where tests are prepared for a required feature or functionality before its corresponding source code is written.
Throwaway Prototype	A full version of a product, intended to be replaced by a subsequent version.
Transition	The phase in the Unified Process model in which the software product is fully delivered and deployed.
Transparency	That everyone can see every aspect of the project.
Triage	A procedure to classify requirements or requests, for example, into “must do”, “should do”, “could do” priorities.

Unified Process Model	A use-case-driven, architecture-focused, iterative and incremental software development process.
Unit Test	A test written and run by developers to verify whether a low-level functionality works correctly.
User Story	A short, structured description of a product requirement that outlines who wants the requirement, what the requirement is, and why the requirement has value.
V-Model	A linear process model, which focuses on verifying the product at multiple levels.
Validated	That the released software product satisfies the client.
Validation	See Validated.
Velocity	The number of units of work that can be completed over a given time interval.
Verification	See Verified.
Verification and Validation Phase	The phase of a software process when the product is tested for whether it meets the requirements and the product is reviewed for whether it satisfies the client.
Verified	That the released software product meets all the specified requirements.
Waterfall Model	A linear software development process, typified by phases where approved work products are passed from one phase to the next.
Work Product	An output produced by completing a specific task. A work product can also be consumed as an input for another task.