

Scrum Software Development Life Cycle

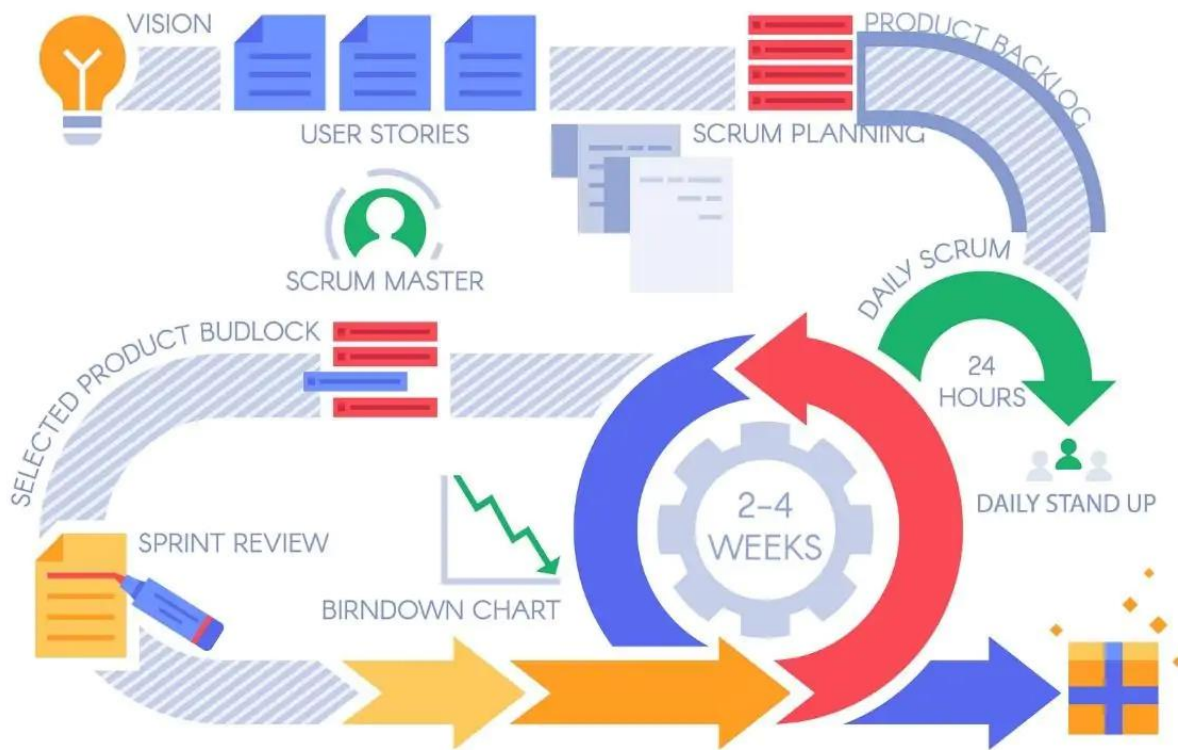


Figure 1: Scrum Software Development Process (Digite, 2022)

In contrast to the waterfall model, Scrum offers an agile methodology in which the development team creates software divided into several phases. In each stage, part of the software will be ready for use as it goes through the steps of planning, building, testing and implementation. Finally, every developed piece of software is delivered to the customer. With the help of customer feedback, the development team can make changes easily as the entire software still needs to be developed. On the contrary, with the waterfall principle, the development team can't make future changes as the whole software is completed when the customer receives the software (Gurendo, 2020).

These are the main roles during the development process, based on the Scrum model:

- The **product owner** is in charge of the end-user's needs.
- The **Scrum master** leads the entire development process. One other function is to ensure the proper use of Scrum and to realise regular Scrum meetings.
- The **Scrum team** develops the product. Its primary tasks are programming, analysis, testing, etc. (Gurendo, 2020)

Software Development Life Circle - Scrum	Processes	Security
Step 1. Product Backlog Creation	<ul style="list-style-type: none"> - Product Owner build a list of ideas and features - Product Owner prioritize the list and brings the top items to the team 	<ul style="list-style-type: none"> - Threat Assessment - Risk Management - Product Hardening (reducing its surface of vulnerability) → Identification of security relevance of items and security planing of next sprint/s)
Step 2: Sprint Planning & Sprint Backlog Creation	<ul style="list-style-type: none"> - Product Owner, Team, Scrum Master discussed the top priority user Stories - User Stories format: As a [User Role], I want to [feature body] so that [User profit] - Determination what can go to next Sprint - Sprint Backlog Creation - Output from Sprint planning meeting - List of User Stories which are committed for the next sprint - Solid understanding who (each Members) is involved 	<ul style="list-style-type: none"> → Planning of security activities and requirements in current sprint
Step 3: Working on the Sprint & Daily	<ul style="list-style-type: none"> - Duration of development: 1 – 3 Weeks' time box 	<ul style="list-style-type: none"> - Risk Management - Product Hardening

Scrum Meetings Sprint	<ul style="list-style-type: none"> - Using a task board to track and set up the to do list - During Sprint (daily stand-up meeting) Discussion of work completion and what the team are working on - Applying burndown charts that allows monitoring daily processes effectively 	<ul style="list-style-type: none"> - Improvement of Security Guidelines, Test & Tools <p>→ Execution and implementation of security activities and requirements, continuous product hardening & identification of security risks</p>
Step 4: Product Increment and Sprint Review	<ul style="list-style-type: none"> - Output: Part of product is potential shippable (After each iteration dev. team creates new version of software product) - Part of the end of every sprint - Product Owner decide if part of product is ready to ship or if there are any additional features needed before shipping 	<ul style="list-style-type: none"> - Risk Management - Product Hardening - Improvement of Security Guidelines, Test & Tools <p>→ Check of existing security dept and security risks & feedback for security backlog</p> <p>→ Comply with security release criteria (e.g. check of existing security risks)</p>
Step 5. Retrospective and Next Sprint Planning	<ul style="list-style-type: none"> - Retrospective review of the results - Discussion on how the development can be improved on the next step - Feedback and reflection of work experiences to improve the teamwork - Planning the next sprint planning 	

Table 1: Scrum Model Step by Step with Security Aspects (Gurendo, 2020; Secodis, 2022)

An example of the task board on Step 3: Working on the Sprint shows the following figure.

















Stories	To Do	In Progress	Testing	Done
 <p>Task #1</p>	   <p>Task #2 Task #3 Task #6</p>	  <p>Task #7 Task #9</p>	 <p>Task #8</p>	  <p>Task #16 Task #17</p>
 <p>New task</p>	  <p>Task #10 Task #11</p>	 <p>Task #12</p>	  <p>Task #13 Task #14</p>	 <p>Task #15</p>

Figure 2: Task board (Gurendo, 2020)

Nowadays, software like Trello takes over the function of a traditional task board. This way, the team can work remotely and keep track of their tasks (Gurendo, 2020).

The following figure shows how to implement security into the agile development life circle. Finally, figure 4 explicitly provides an excellent overview of how security is managed in each step. These are also indicated in the table above.

Security in the Agile Lifecycle

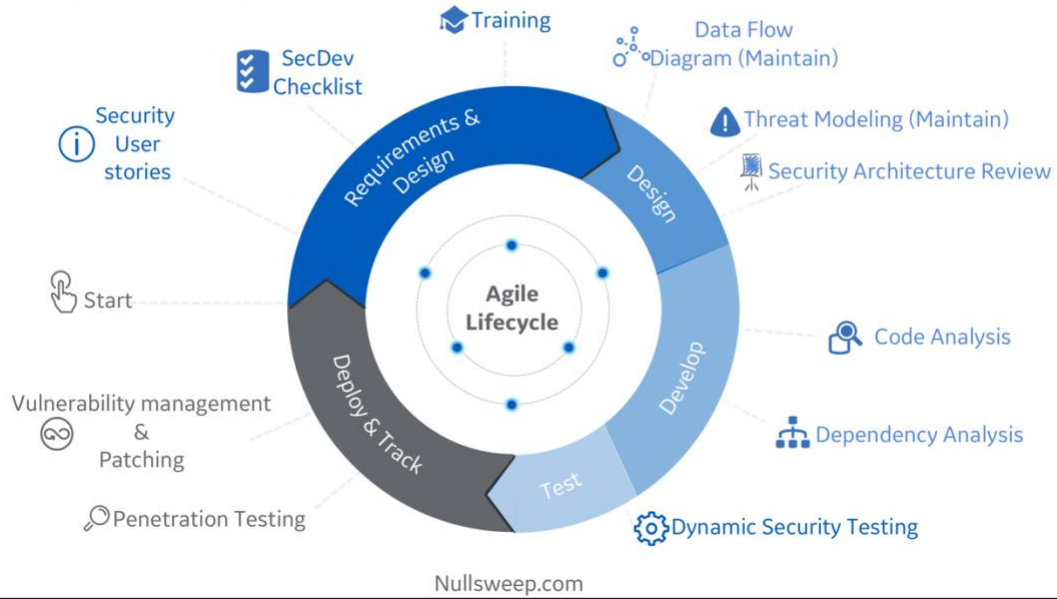


Figure 3: Secure Software Development Life Circle (Nullsweep, 2019)

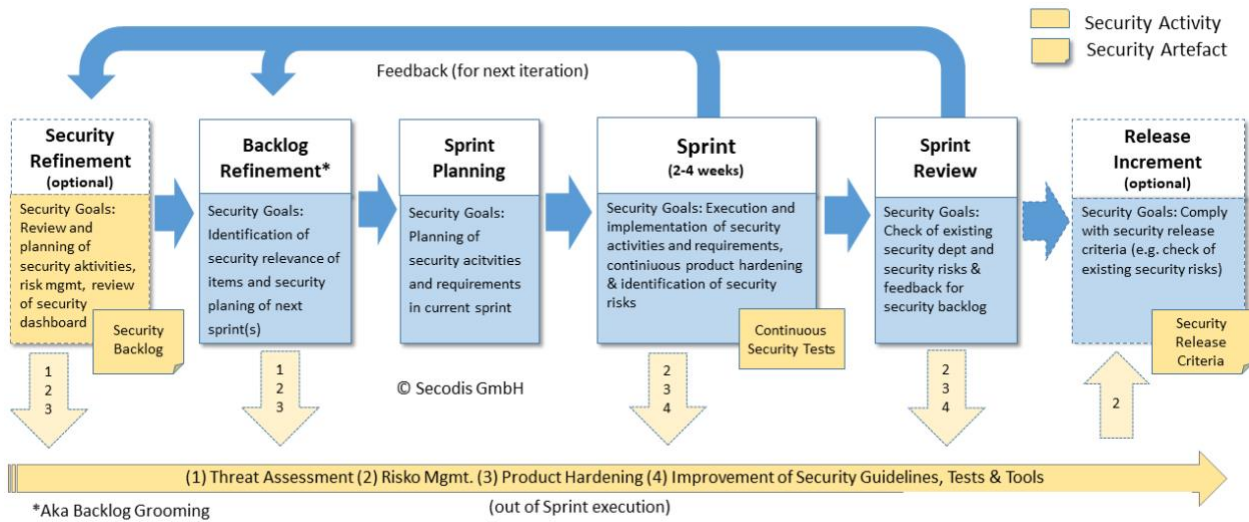


Figure 4: Secure Software Development with Scrum (Secodis, 2022)

References

Digite (2022) What Is Scrum Methodology? & Scrum Project Management. Available from: <https://www.digite.com/agile/scrum-methodology/> [Accessed 29 September 2022].

Gurendo, D. (2020) Software Development Life Cycle (SDLC). Scrum Model Step by Step. Available from: <https://xbsoftware.com/blog/software-development-life-cycle-sdlc-scrum-step-step/> [Accessed 29 September 2022].

Nullsweep (2019) Integrating Security With Agile Development. Available from: <https://nullsweep.com/integrating-security-with-agile-development/> [Accessed 29 September 2022].

Secodis (2022) Agile Security. Available from: <https://www.secodis.com/agile-security/?lang=en> [Accessed 29 September 2022].