**SCRUM MODEL:**

Scrum is an agile project management methodology or framework used primarily for software development projects with the goal of delivering new software capability every 2-4 weeks. It is one of the approaches which articulates a set of values and principles to guide decisions on how to develop higher-quality software faster. Scrum is widely used by functions including IT and marketing where there are projects that must move forward in the presence of complexity and ambiguity.

Scrum model have certain advantages or special benefits like:

* Higher productivity
* Better quality products
* Reduced time to market
* Better team dynamics
* Improved stakeholder satisfaction.

Scrum addresses complexity in work by making information transparent, so that people can inspect and adapt based on current conditions, rather than predicted conditions. This allows teams to address the common pitfalls like chaos resulting from constantly changing requirements; underestimation of time, resources and cost; compromises on software quality; and inaccurate progress reporting. Transparency of common terms and standards is required in Scrum development to ensure that what is being delivered is what was expected. Frequent inspection ensures progress and detects variances early on so that adjustments can be made quickly.

The Scrum methodology is defined by team roles, events, artifacts, and rules.

Scrum teams are typically composed of 5-7 members and have no team leader to delegate tasks or decide how a problem is solved. The team as a unit decides how to address issues and solve problems. Each member of the Scrum team is an integral part of the solution and is expected to carry a product from inception to completion. There are three key roles in a Scrum team:

The product owner:

The Product Owner is the project's key stakeholder - usually an internal or external customer, or a spokesperson for the customer. There is only one Product Owner who conveys the overall mission and vision of the product which the team is building. The Product Owner is ultimately accountable for managing the product backlog and accepting completed increments of work.

The scrum master:

The scrum master is the servant leader to the Product Owner, Development Team and Organization. With no hierarchy authority over the team but rather more of a facilitator, the ScrumMaster ensures that the team adheres to Scrum theory, practices, and rules. The ScrumMaster protects the team by doing anything possible to help the team perform at the highest level. This may include removing impediments, facilitating meetings, and helping the Product Owner groom the backlog.

The development team:

The Development Team is a self-organizing, cross-functional group armed with all the skills to deliver shippable increments at the completion of each sprint. Scrum broadens the definition of the term "developer" beyond programmers to include anyone who participates in the creation of the delivered increment. There are no titles in the Development Team and no one, including the ScrumMaster, tells the Development Team how to turn product backlog items into potentially shippable increments.

**Scrum events:**

The sprint:

A sprint is a time-boxed period during which specific work is completed and made ready for review. Sprints are usually 2-4 weeks long but can be as short as one week.

Sprint planning:

Planning team meetings are time-boxed events that determine which product backlog items will be delivered and how the work will be achieved.

The daily stand up:

The Daily Stand-up is a short communication meeting (no more than 15 minutes) in which each team member quickly and transparently covers progress since the last stand-up, planned work before the next meeting, and any impediments that may be blocking his or her progress.

The sprint review:

The Sprint Review is the "show-and-tell" or demonstration event for the team to present the work completed during the sprint. The Product Owner checks the work against pre-defined acceptance criteria and either accepts or rejects the work. The stakeholders or clients give feedback to ensure that the delivered increment met the business need.

The retrospective:

The Retrospective, or Retro, is the final team meeting in the Sprint to determine what went well, what didn't go well, and how the team can improve in the next Sprint. Attended by the team and the ScrumMaster, the Retrospective is an important opportunity for the team to focus on its overall performance and identify strategies for continuous improvement on its processes.

Scrum artifacts:

Product backlog:

The product backlog is the single most important document that outlines every requirement for a system, project or product. The product backlog can be thought of as a to-do list consisting of work items, each of which produces a deliverable with business value. Backlog items are ordered in terms of business value by the Product Owner.

Sprint backlog:

A sprint backlog is the specific list of items taken from the product backlog which are to be completed in a sprint.

Increment:

An Increment is the sum of all product backlog items that have been completed since the last software release. While it is up to the Product Owner to decide on when an increment is released, it is the team's responsibility to make sure everything that is included in an increment is ready to be released. This is also referred to as the Potentially Shippable Increment (PSI).

Scrum Rules

The rules of agile Scrum should be completely up to the team and governed by what works best for their processes. The best agile coaches will tell teams to start with the basic scrum events listed above and then inspect and adapt based on your team's unique needs so there is continuous improvement in the way teams work together.

Practicing Scrum

Getting started:

It is not uncommon for an individual Scrum team to use simple Scrum tools like a whiteboard, sticky notes, or a spreadsheet to manage the product backlog and the progress of the sprint backlog items in each sprint. Scaling agile practices to the rest of the organization is undoubtedly more complicated: The more teams use Scrum within an organization or are geographically dispersed, the more cumbersome simple tools like whiteboards, sticky notes, and spreadsheets become.

We can implement this in our game team by considering our each teams as one of a scrum teams where we can consider the team leaders as the scrum master of each scrum team. And all the teams will be under a project manager. All teams will follow sprint, sprint planning, sprint review, product backlog etc and implement their work together as a whole team.

**AGILE MODEL:**

Agile model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product. Agile model believes that every project needs to be handled differently and the existing methods need to be tailored to best suit the project requirements. In Agile, the tasks are divided to small time frames to deliver specific features for a release.

Iterative approach is taken and working software build is delivered after each iteration. Each build is incremental in terms of features; the final build holds all the features required by the customer.

Agile follows such principles as responding to change, customer collaboration, interactions like co-location and pair programming, demo working software.

Agile Methods break the product into small incremental builds. These builds are provided in iterations. Each iteration typically lasts from about one to three weeks. Every iteration involves cross functional teams working simultaneously on various areas like −

* Planning
* Requirements Analysis
* Design
* Coding
* Unit Testing and
* Acceptance Testing.

At the end of the iteration, a working product is displayed to the customer and important stakeholders.

Agile uses an adaptive approach where there is no detailed planning and there is clarity on future tasks only in respect of what features need to be developed. There is feature driven development and the team adapts to the changing product requirements dynamically. The product is tested very frequently, through the release iterations, minimizing the risk of any major failures in future.

Customer Interaction is the backbone of this Agile methodology, and open communication with minimum documentation are the typical features of Agile development environment. The agile teams work in close collaboration with each other and are most often located in the same geographical location.

The advantages of the Agile Model are as follows – It Is a very realistic approach to software development, Promotes teamwork and cross training, Functionality can be developed rapidly and demonstrated, Resource requirements are minimum, Suitable for fixed or changing requirements, Delivers early partial working solutions, Good model for environments that change steadily, Minimal rules, documentation easily employed, Enables concurrent development and delivery within an overall planned context, Little or no planning required, Easy to manage.

But there are some issues also.

* Not suitable for handling complex dependencies.
* More risk of sustainability, maintainability and extensibility.
* An overall plan, an agile leader and agile PM practice is a must without which it will not work.
* Strict delivery management dictates the scope, functionality to be delivered, and adjustments to meet the deadlines.
* Depends heavily on customer interaction, so if customer is not clear, team can be driven in the wrong direction. There is a very high individual dependency, since there is minimum documentation generated.
* Transfer of technology to new team members may be quite challenging due to lack of documentation.

If we use this development model in our team, we can implement this model by dividing tasks of the iterative stages to our different teams and complete the steps of the iterative processes altogether. We can solve the disadvantage issues by having an overall plan layout, having the project head as an agile leader and by having clear customer interactions with each team.

**GOOGLE ADSENSE:**

Google AdSense is a free, simple way to earn money by displaying ads next to our online content. With AdSense, we can show relevant and engaging ads to our app or game visitors and even customize the look and feel of ads to match our game. The ads are created and paid for by advertisers who want to promote their products. Since these advertisers pay different prices for different ads, the amount we earn will vary.

The AdSense Program differs in that it delivers ads served by Google Ads to our game. Google then pays us for the ads displayed on our game based on user clicks on ads or on ad impressions, depending on the type of ad. AdSense gives us instant and automatic access to a huge source of advertiser demand, which means competition for our ad spaces, more relevant ads, and ads for all our online content. AdSense uses an ad auction to automatically select the ads that will appear on our game. The highest paying ads will show on our game.

For signing up to google AdSense, we need to add an url website and an email address, then we are signed up in google AdSense and we can use it by following its program policies. Participation in AdSense is free. Even better, Google will pay us for clicks, impressions, and other interactions with the Google ads we'll display on our game.

There are three steps:

|  |  |
| --- | --- |
| * We make our ad spaces available by pasting ad code on our game, and choose where we want the ads to appear. * Advertisers bid to show in our ad spaces in a real-time auction. The highest paying ads show on our game. * Google handle the process of billing all advertisers and networks for the ads on our game, to make sure we receive our payments.   There are few types of ads we can use in our game:    BANNER:  Banner ads are rectangular image or text ads that occupy a spot within an app's layout. They stay on screen while users are interacting with the app, and can refresh automatically after a certain period of time. If you're new to mobile advertising, they're a great place to start.  INTERSTITIAL:  Interstitials are full-screen ads that cover the interface of an app until closed by the user. They're best used at natural pauses in the flow of an app's execution, such as in between levels of a game or just after completing a task.  NATIVE:  Native is a component-based ad format that gives the freedom to customize the way assets such as headlines and calls to action are presented in the apps. By choosing fonts, colors, and other details, we can create natural, unobtrusive ad presentations that can add to a rich user experience.  REWARDED:  Rewarded video ads are full-screen video ads that users have the option of watching in full in exchange for in-app rewards.  **HOW TO EARN FROM GOOGLE ADSENSE:**  There are 2-3 methods which we can follow. The most used method has certain steps which we need to do for earn money from it.   * Creating an ad unit, * Sign in to the adsense account and name the ad unit, * Pick a size, * Set the ad type, * Create a custom channel to check the performance, * Create the ad style like the various components of an ad: border,title,background,text etc. * Get the html code for the ad lastly. |  |