|  |  |
| --- | --- |
| Advantage | * Agile scrum helps the company in saving time and money. * [Scrum methodology](http://www.my-project-management-expert.com/scrum-methodology.html) enables project’s where the[business requirements documentation](http://www.my-project-management-expert.com/business-requirements-documentation.html) is hard to quantify to be successfully developed. * Fast moving, cutting edge developments can be quickly coded and tested using this method, as a mistake can be easily rectified. * It is a lightly controlled method which insists on frequent updating of the progress in work through regular meetings. Thus there is clear visibility of the project development. * Like any other [agile methodology](http://www.my-project-management-expert.com/agile-methodology.html), this is also iterative in nature. It requires continuous feedback from the user. * Due to short sprints and constant feedback, it becomes easier to cope with the changes. * Daily meetings make it possible to measure individual productivity. This leads to the improvement in the productivity of each of the team members. * Issues are identified well in advance through the daily meetings and hence can be resolved in speedily * It is easier to deliver a quality product in a scheduled time. * Agile Scrum can work with any technology/ programming language but is particularly useful for fast moving web 2.0 or new media projects. * The overhead cost in terms of process and management is minimal thus leading to a quicker, cheaper result |
| Disadvantage | * Agile Scrum is one of the leading [causes of scope creep](http://www.my-project-management-expert.com/causes-of-scope-creep.html) because unless there is a definite end date, the [project management stakeholders](http://www.my-project-management-expert.com/project-management-stakeholders.html) will be tempted to keep demanding new functionality is delivered. * If a task is not well defined, [estimating project costs](http://www.my-project-management-expert.com/estimating-project-costs.html)and time will not be accurate. In such a case, the task can be spread over several sprints. * If the team members are not committed, the project will either never complete or fail. * It is good for small, fast moving projects as it works well only with small team. * This methodology needs experienced team members only. If the team consists of people who are novices, the project cannot be completed in time. * Scrum works well when the Scrum Master trusts the team they are managing. If they practice too strict control over the team members, it can be extremely frustrating for them, leading to demoralisation and the failure of the project. * If any of the team members leave during a development it can have a huge inverse effect on the project development * [Project quality management](http://www.my-project-management-expert.com/project-quality-management.html) is hard to implement and quantify unless the test team are able to conduct regression testing after each sprint. |

- how to divide and coordinate the

tasks so that everything can be done without impediments

And one things . (turn left) he said that , but does not mean you understand right. Even you give a example

|  |  |  |
| --- | --- | --- |
|  | Something | Scrum |
| **PROCESS** |  |  |
| Quality | Quality focus changes from Analysis > Design > Code > Test | Quality focus on all aspects of SDLC at any given time |
| Quality control | Detection & fixing during system and regression testing at the last phase of project | Early detection & fixing in each sprint followed by stabilization |
| CA & PA | Lessons learned from the previous release implemented in release | Lessons learned from previous sprint implemented in next sprint |
| Risk | No Risk identification & Firefighting during testing phase | Early identification & mitigation in every sprint |
| Postmoterm | After every release | After every sprint in retrospection |
| Customer feedback | At the end of the project | At the end of every sprint |
| **PLANNING** |  |  |
| Definition of ready | Analysis and design should be completed for all stories before programming | Stories that are not subjected to change , that can be completed  Within the sprint will be considered for the sprint |
| Scoping | Product Owner decides project scope | Team decides the sprint scope as per capacity availability & product backlog prioritization |
| Effort Estimation | PM provides estimates and get approval from PO for entire project | Scrum Master facilitates and Team does the estimation. Story points can be reviewed and refined during sprint planning meeting |
| Plan Review | Team need to stick to baselined project plan | Team can review during mid-sprint planning |
| **EXECUTION** |  |  |
| Goals | Goals are defined for each phase by defining entry and exit criteria. Delivering artifacts | Completing the feature/story in all aspects within a sprint .Delivering the shippable product. |
| Phase silos or roles silos or story silos | Analysis and design will be completed for all stories before proceeding to coding and testing | Team gets divided into mini teams and each team focus on story completion in all aspects. |
| Resource utilization | Role specific. Resources restricted to the tasks that suits/matches their role only. | Everybody is ready to work any task so as to complete the story |
| Ownership | Ownership changes from role to role by phase to phase. Roles play key role. PM responsible for overall delivery | Entire team is responsible for story completion in all aspects – analysis , design , development , testing and demo |
| Task Assignment | Project manager do the feature/task assignment for entire project | Team members are empowered to own the feature/tasks for every sprint |
| Daily standup | Daily/Weekly Statues reports. PM calls for the status meeting at scheduled intervals | Scrum Master facilitates the daily standup meeting: what we did?  What we are going to do  And where we lag? |
| Statius report | Status report in prescribed template more focus on Percentage done | Update version one on daily basis reflecting actual hours. Burn down chart measures |
| Planned vs actual | Stick to planed baselined schedule the project | Update the status with actual burn down hours |
| Testing resources | Only testers will identify , prepare , and execute the TCs | Anyone in the team identify , prepare and execute TCs |
| New feature testing | TC preparation and execution starts after analysis , design and development phase | TC preparation and execution starts in parallel with analysis and design tasks and completes within same sprint |
| *Regression testing* | *After the completion of first cycle system testing* | *Full coverage within the sprint* |
| Stabilization phase | NA | Starts after the completion of development sprints No feature Development |
| **COMPLENTION** |  |  |
| Definition of done | DOD for a story will be measured by splitting story into analysis , design , development and testing deliverables , which are validated , reviewed | DOD for a story will be measured by its capability of demonstration and delivery to the external world |
| Delivery | Delivery at the end of the project | Demonstration and delivery at the end of every sprint |
| Story closing | Delivering/Publishing artifacts(DOD) phase by phase | Closing the story based on demo feedback, test case results and defect status. |

The prove the points form the above chart we would

|  |
| --- |
| Quality |
| Quality control |
| CA & PA |
| Risk |
| Postmoterm |
| Customer feedback |

Is true because Since the waterfall is … is design at the last stage, so there will keep unknown until that stage has been reach.

|  |
| --- |
| Definition of ready |
| Scoping |
| Effort Estimation |
| Plan Review |

Also make scene because the design phrase in waterfall must be done before starting type a line of code but for the SCRUM will just need the design the part that we are going the impenment , and then design the next part after we finish the pervious part

|  |
| --- |
| Goals |
| Phase silos or roles silos or story silos |
| Resource utilization |
| Ownership |
| Task Assignment |
| Daily standup |
| Statius report |
| Planned vs actual |
| Testing resources |
| New feature testing |
| *Regression testing* |
| Stabilization phase |

For the goals

|  |
| --- |
| Definition of done |
| Delivery |
| Story closing |

At the end

After we looked at the difference between them , we would be interesting to have a look at actually how many project have been sceccus in waterfall and SCRUM

As the picture above we can see …

From the analysis from the information we can clearly see that compare the waterfall model the SCRUM is more suitable for the game industry , the reason why is

* Customers always change
* A lot of factor can be effecting to build the project and usually some big problem does not come until some time
* We know we want a fun game but it is hard to define what is fun, so there must be a lot of play testing and feedback from the user to make a fun
* It gives motivation if the team can see something, because they can trust it will be finish.
* From these condition above we can see SCRUM is more suitable then waterfall model.

## Conclusion

Compare the waterfall and SCRUM model , The SCRUM model is more suitable in game industry because

.

http://sbgames.org/sbgames2010/proceedings/computing/short/Computing\_short19.pdf

# Conclusion

Clearly that we can see compare the waterfall and scrum , scrum is suitable for in game industry