|  |  |
| --- | --- |
| A picture of a winding road and trees  Development Life Cycles  [Document subtitle] | Abstract  [Draw your reader in with an engaging abstract. It is typically a short summary of the document. When you’re ready to add your content, just click here and start typing.]  kitman yiu  [Course title] |

content

[Introduction 2](#_Toc396652268)

[SDLC 3](#_Toc396652269)

[Introduction 3](#_Toc396652270)

[Content 3](#_Toc396652271)

[Conclusion 3](#_Toc396652272)

[Waterfall 5](#_Toc396652273)

[Introduction 5](#_Toc396652274)

[Content 5](#_Toc396652275)

[Example of using waterfall 6](#_Toc396652276)

[Conclusion 6](#_Toc396652277)

[SCRUM 8](#_Toc396652278)

[Introduction 8](#_Toc396652279)

[Content 9](#_Toc396652280)

[Product Owner 9](#_Toc396652281)

[Scrum Master 9](#_Toc396652282)

[Team 9](#_Toc396652283)

[The sprint meeting 12](#_Toc396652284)

[Diaily Scrum 12](#_Toc396652285)

[Sprint retrospective meeting 12](#_Toc396652286)

[Example of using SCRUM 13](#_Toc396652287)

[Conclusion 13](#_Toc396652288)

[Difference between agile and waterfall 15](#_Toc396652289)

[Content 15](#_Toc396652290)

[Conclusion 18](#_Toc396652291)

[Conclusion 18](#_Toc396652292)

[Reference 19](#_Toc396652293)

# Introduction

In game industry when we are creating a big project we would want to manage our project. And in game industry the things that helps us the manage out project is known as SDLC(software development life cycles). And Inside SDLC there are a lot method, so I pick two inside SDLC and the named are called SCRUM and waterfall. In this article I will talk about what is SDLC , and also what is SCRUM and waterfall and at the end which is the main discussion part , which we will take a look the difference between SCRUM and waterfall. In this article it is recommend to read the conclusion first and then look at the example after, if you still do not understand the paragraph then we recommend to read the content part.

# 

# SDLC

## Introduction

In this paragraph, I will write about what is where (SDLC) software development life cycles, why we use SDLC, the advantage of SDLC and have a look different types of SLDC.

## Content

**(Wikipedia: System development life cycle)**SDLC(software development life cycles) is a subset of software development methodology , SDLC is a term to describe a process for planning , creating , testing and deploying an information system. The SDLC can be use is applies a process to a range of software and hardware configurations as a system can be composed of hardware only , software only , or a combination of both .The reason of using SDLC is because it provides basic for project planning , estimating & scheduling , and also provides mechanism for project tracking & control and all these leads to some advantage for example, increase development speed , increase product quality , improve tracking & control , improve client relations , decrease project risk and decreased project management overhead all these end up to help the company to save time and earn more money. There are many methodology that has been discovered since ,…. .The following focal points are listed below:

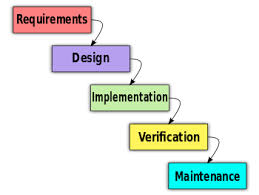
* [Waterfall](http://en.wikipedia.org/wiki/Waterfall_model)
* [Prototype model](http://en.wikipedia.org/wiki/Software_prototyping)
* [Incremental](http://en.wikipedia.org/wiki/Incremental_build_model)
* …….

So for this … I choose the talk about waterfall and scrum methodology

## Conclusion

|  |  |
| --- | --- |
| **What is SDLC** | Is a term to describe a process for planning, creating, testing and deploying an information system |
| **Who will use SDLC** | This term is used in system engineering, information system and software engineering |
| **Where to use SDLC** | The SDLC is applices a process to a range of software and hardware configurations, as a system can be composed of hardware only , software only , or a combination of both . |
| **Advantage** | * Although there are SDLC models that could cater to strict time pressures, program development is faster when programmers come together to create the workflow instead of going through a lot of user inputs and requirements. * Another advantage is a clear purpose on why is the software created. Developers will just have to work on the problem and create a solution based on the need. SDLC will have so many people involve there is a tendency that the real purpose of creating the software will be compromised. |
| **Disadvantage** | The closest disadvantage anyone could think of SDLC is the difference between what is written in paper and what is actually implemented. There are things that are happening in the actual work that the paper doesn’t see. This gives a good impression for the clients especially for 3rd party developers but when the software is actually launched it’s on a very bad situation. The actual situation of software development could be covered by fancy paperwork of SDLC.  Another disadvantage of a program or software that follows the SDLC program is it encourages stiff implementation instead of pushing for creativity in different software. |
| Types of SDLC | * [Waterfall](http://en.wikipedia.org/wiki/Waterfall_model) * [Prototype model](http://en.wikipedia.org/wiki/Software_prototyping) * [Incremental](http://en.wikipedia.org/wiki/Incremental_build_model) * [Iterative](http://en.wikipedia.org/wiki/Iterative_and_incremental_development) * [V-Model](http://en.wikipedia.org/wiki/V-Model_(software_development)) * [Spiral](http://en.wikipedia.org/wiki/Spiral_model) * [Scrum](http://en.wikipedia.org/wiki/Scrum_(software_development)) * [Cleanroom](http://en.wikipedia.org/wiki/Cleanroom_software_engineering) * [RAD](http://en.wikipedia.org/wiki/Rapid_application_development) * [DSDM](http://en.wikipedia.org/wiki/Dynamic_systems_development_method) * [UP](http://en.wikipedia.org/wiki/Unified_Process) * [XP](http://en.wikipedia.org/wiki/Extreme_programming) * [Agile](http://en.wikipedia.org/wiki/Agile_software_development) * [Lean](http://en.wikipedia.org/wiki/Lean_software_development) * [Dual Vee Model](http://en.wikipedia.org/wiki/Dual_Vee_Model) * [TDD](http://en.wikipedia.org/wiki/Test-driven_development) * [BDD](http://en.wikipedia.org/wiki/Behavior-driven_development) * [FDD](http://en.wikipedia.org/wiki/Feature-driven_development) * [DDD](http://en.wikipedia.org/wiki/Domain-driven_design) * [MDD](http://en.wikipedia.org/wiki/Model-driven_development) |

# Waterfall



## Introduction

In this paragraph I will explain a little bit of the waterfall model , after that I will write about what is waterfall and how it works , and also I will write about the advantage and disadvantage of the waterfall model , and at the end of this paragraph I will give an example how the waterfall model works

## Content

**(Wikipedia:Waterfall model)**The waterfall model is one of model which used in software development process, The first formal description of the waterfall model is often cited as a 1970 article by [Winston W. Royce](http://en.wikipedia.org/wiki/Winston_W._Royce) . although Royce did not use the term "waterfall" in that article. Royce presented this model as an example of a flawed, non-working model. This, in fact, is how the term is generally used in writing about software development—to describe a critical view of a commonly used software development practice.

**(Tutorialspoint)**It is called as such because the model develops systematically from one phase to another phase in a downward fashion, like a waterfall. In waterfall model there are five phase include the Requirements, Design, Implements, Verification, Maintenance. The requirement are all possible requirements of the system to be developed are captured in this phase and also all the document that need to be made in this step. The design are to study the requirement specifications from last phase and to prepare the system design. System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture. The Implements are inputs from system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality which is referred to as Unit Testing. The Verification are all the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures. The Maintenance are some issues which come up in the client environment. To fix those issues patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

**(Tutorialspoint)**The advantage of using waterfall model is the project phase must be fully finish before moving onto another phase. Therefore if there is a fault in this software it will be detected during one of the initial phases and will be sealed off for correction, also a lot of emphasis is laid on paperwork in this method as compared to the newer methods. When new workers enter the project, it is easier for them to carry on the work from where it had been left. The newer methods don’t document their developmental process which makes it difficult for a newer member of the team to understand what step is going to follow next. The Waterfall Model is a straight forward method and lets one know easily what stage is in progress, the last advantage is the Waterfall method is also well known amongst the software developers therefore it is easy to use. It is easier to develop various software through this method in short span of time.

**(Tutorialspoint)**After we look at the advantage we will look at the disadvantage of waterfall model, one of the disadvantage of waterfall model is many project has a lot of external factors, one of them are clients, because clients used the change there requirements a lot time. But since how waterfall is design it doesn’t work well in a situation like this as it assumes no alteration to occur once the process has started according to plan , so what happens is then a number of steps would go to waste, and there would arise a need to start everything all over again. Of course this also brings about the aspect of time and money which will all go to waste. Therefore this method will not at all prove to be cost effective. It is not even easy to take out the cost estimate of each step, as each of the phases is quite big. Another disadvantage is this model is that a huge amount of time is also wasted. For example if we study any software development process, we know that Phase II cannot be executed until Phase I has been successfully completed; so while the designers are still designing the software, time of the builders is completely wasted. And as the picture above we see because the testing phrase comes quite late in the development process whereas in various other developmental programs the designs would be tested a lot sooner to find the flaw at a time when a lot of time and money has not been wasted. And the last is Elaborate documentation during the Waterfall method has its advantages, but it is not without the disadvantages as well. It takes a lot of effort and time, which is not suitable for smaller projects.

However this model is not used in game industry at all, which will explain why in the next chapter, the waterfall are good for **(see : Waterfall software and project development model)**military projects, aerospace projects and similar high-responsibility software. Because these project cannot take any risk at the end so all the requirements must be collected before it start the implement the code. And also it can’t be test fail.

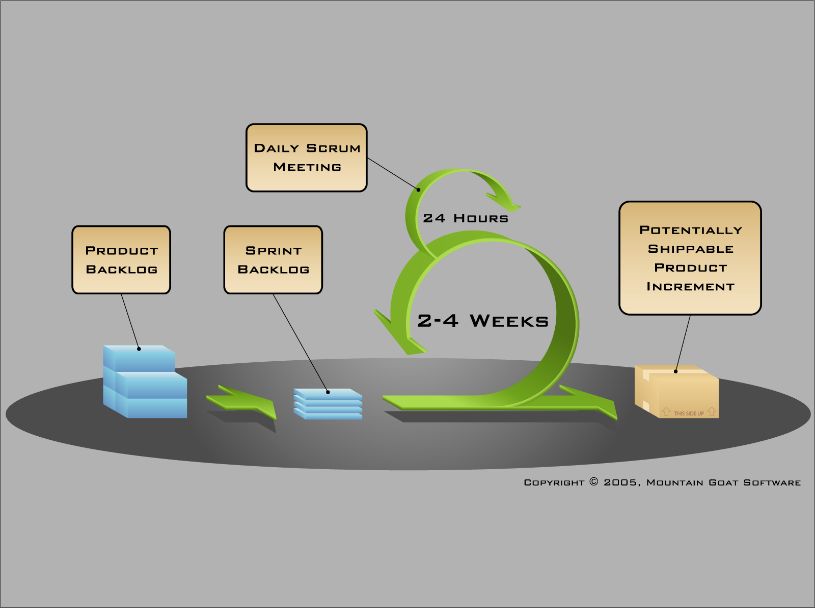
## Example of using waterfall

Shooting a rocket…

## Conclusion

|  |  |
| --- | --- |
| What is waterfall | A model which used in software development process, systematically from one phase to another phase in a downward fashion, like a waterfall. In waterfall model there are five phase |
| Different phase of waterfall | * **Requirement Gathering and analysis:** All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification doc. * **System Design:** The requirement specifications from first phase are studied in this phase and system design is prepared. System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture. * **Implementation:** With inputs from system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality which is referred to as Unit Testing. * **Integration and Testing:** All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures. * **Deployment of system:** Once the functional and non-functional testing is done, the product is deployed in the customer environment or released into the market. * **Maintenance:** There are some issues which come up in the client environment. To fix those issues patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment. |
| Advantage | * Simple and easy to understand and use * Easy to manage due to the rigidity of the model. Each phase has specific deliverables and a review process. * Phases are processed and completed one at a time. * Works well for smaller projects where requirements are very well understood. * Clearly defined stages. * Well understood milestones. * Easy to arrange tasks. * Process and results are well documented. |
| Disadvantage | * High amounts of risk and uncertainty. * Not a good model for complex and object-oriented projects. * Poor model for long and ongoing projects. * Not suitable for the projects where requirements are at a moderate to high risk of changing. So risk and uncertainty is high with this process model. * It is difficult to measure progress within stages. * Cannot accommodate changing requirements. * No working software is produced until late in the life cycle. * Adjusting scope during the life cycle can end a project. * Integration is done as a "big-bang. at the very end, which doesn't allow identifying any technological or business bottleneck or challenges early. * No working software is produced until late during the life cycle. |
| Where has been used | * Military projects, * Aerospace projects * Similar high-responsibility software |

# SCRUM



## Introduction

Same as above the paragraph , I will first talk about the history of Scrum and how does scrum work or in other word what element does scrum have after that I will talk about the advantage and disadvantage of Scrum , and of course at the end give an example that how does scrum work.

## Content

**(Wikipedia:Scrum(software development))** Scrum is an iterative and incremental agile software development framework for managing product development.

Scrum was first defined as "a flexible, holistic product development strategy where a development team works as a unit to reach a common goal" as opposed to a "traditional, sequential approach" in 1986 by [Hirotaka Takeuchi](http://en.wikipedia.org/wiki/Hirotaka_Takeuchi" \o "Hirotaka Takeuchi) and [Ikujiro Nonaka](http://en.wikipedia.org/wiki/Ikujiro_Nonaka" \o "Ikujiro Nonaka) in the "New New Product Development Game". Takeuchi and Nonaka later argued in "The Knowledge Creating Company" that it is a form of "organizational knowledge creation, [...] especially good at bringing about innovation continuously, incrementally and spirally".

(See Video:NEW Intro to Agile Scrum in Under 10 Minutes - What is Scrum?) In Scrum there are three few elements: roles, meetings, artifacts, and the different roles are: Product Owner, Team, and [Scrum Master](http://scrummasterchecklist.org/pdf/ScrumMaster_Checklist_12_unbranded.pdf).

### Product Owner

The Product Owner responsibility is to remove any [impediments](http://scrummethodology.com/scrum-impediments/) that obstruct a team’s pursuit of its sprint goals. It is important that because its job is to make sure the right features make it into the product backlog. Representing the users and customers of the product. She helps set the direction of the product

### [Scrum Master](http://scrummasterchecklist.org/pdf/ScrumMaster_Checklist_12_unbranded.pdf)

After the Product Owner we will take a look at the Scrum Master. It job is responsible for communicating the vision of the product to the development team. He or she must also represent the customer’s interests through requirements and prioritization. Because the Product Owner has the most authority of the three roles, it’s also the role with the most responsibility. The Product Owner is the single individual who must face the music when a project goes awry. Also Scrum Master needs to help the [Product Owner](http://scrummethodology.com/scrum-product-owner/) maximize productivity or helping the team turn the [sprint retrospective meeting](http://scrummethodology.com/scrum-meetings) into an evolutionary / [Kaizen](http://en.wikipedia.org/wiki/Kaizen)experience. Facilitating for the team or [Product Owner](http://scrummethodology.com/scrum-product-owner/) might also include tasks like helping maintain the [backlog](http://scrummethodology.com/the-scrum-backlog/) and release plan or radiating Scrum artifacts to ensure the [Product Owner](http://scrummethodology.com/scrum-product-owner/) or The Team is informed about progress. See more from http://www.scrummasterchecklist.org/pdf/ScrumMaster\_Checklist\_12\_unbranded.pdf

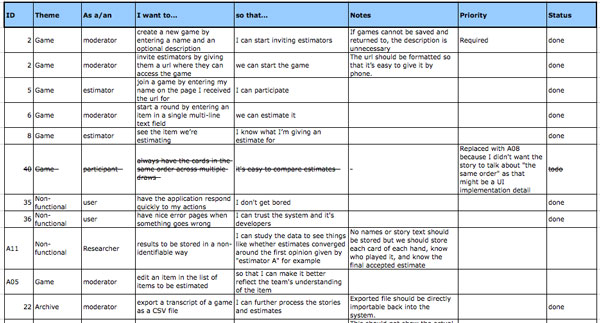
### 

### Team

And the last role in SCRUM is the team, the team is responsible for completing work. Ideally, teams consist of seven cross-functional members, plus or minus two individuals. For software projects, a typical team includes a mix of software engineers, architects, programmers, analysts, QA experts, testers, and UI designers. Each sprint, the team is responsible for determining how it will accomplish the work to be completed. This grants teams a great deal of autonomy, but, similar to the Product Owner’s situation, that freedom is accompanied by a responsibility to meet the goals of the sprint.

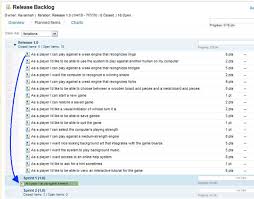
The different artifacts of SCRUM are Product Backlog, Sprint Backlog, Burndown chart, Release Backlog

### Product Backlog



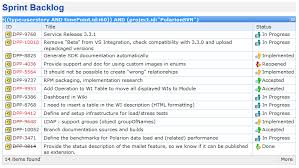
The collection of all these user-stories is called product backlog. The user-stories are kind of feature requests from the customers, executives, or even other team members that have been written from the perspective of the end –user. And the recommend format is As a (role) , I want (feature) , so that (benefit).Or another way to think of a product backlog is a wish list of all the things that would make this product great.

### Release Backlog



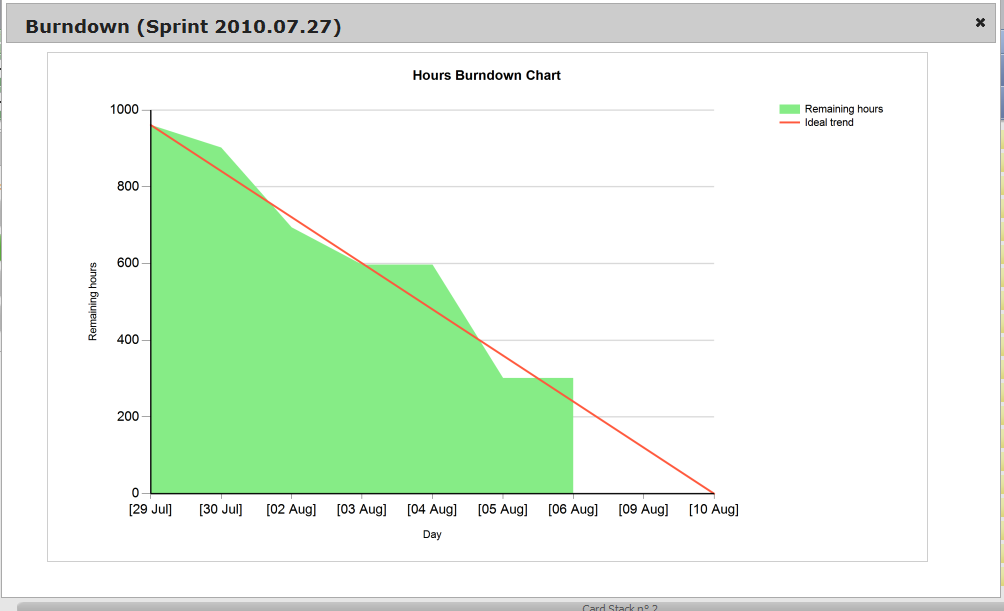
The product owner will identify the user-stories they want to put into this release. After he decided what he want to put in the release he will put it into the release backlog. The team then prioritizes the user-stories and estimates the amount of work involved for each item. Sometimes larger user-stories are broken down into smaller manageable chucks. The Collection of all the estimates provides a rough idea of the total amount of work involved to complete the entire release.

### Sprint Backlog



Which take from the release backlog and split it up into several of these, One of the most important things remember about sprints is that the goal each sprint is to get a subset of the release backlog to a ship-ready state ,so at the end of each sprint , you should have a fully tested product with all the features of that sprint 100% complete Since sprints are a very short , but a realistic representation of part of the product, a late finish of the sprint is a great indicator that the project is not on schedult and something needs to be done. Therefore, it’s extremely important to monitor the progress of each progress of each sprint with the next chart called burndown chart.

### Burndown chart



The burndown chart is the number one reason for Scrum’s popularity, because this ensure a project is progressing smoothly by this visibility tool. The burndown chart provides a day-by-day measure of amount of work that remains in a given sprint or release. The burndown chart provides a day-by-day measure of the amount of work that remains in a given sprint or release, From the graph above , you can see that the amount of work remaining bounces up and down from day to day but is generally trending towards zero. Because historical information is provided in the burndown chart, it’s easy for us to see if the team is on the right track, using the burndown chart the team can quickly calculate the slope of the graph which is also called the burndown Velocity this is the average of productivity for each day. For example, a team’s rate of productivity might be that on typical day they finish approximately 50 hours of work. Knowing that, it’s possible to calculate estimated completion date for the sprint or even for the entire release, based on the amount of work remaining. Knowing that, it’s possible to calculate an estimated completion date for the sprint or even for the entire release, based on the amount of work remaining. The great things about the burn down chart is that we can compare our actual velocity and projected completion date to what the team needs to do in order to finish OnTime. Because knowing this can help makes the proper adjustments necessary to get the project on track. This will show us is the project going to be on time or be late.

# The different meetings of SCRUM are Sprint Planning, Daily Scrum, and Retrospective

### The sprint meeting

Every iteration starts with a sprint planning meeting. The product owner holds a conversation with the team and decides which stories are highest in priority, and which ones they will tackle first. Stories are added to the sprint backlog, and the team then breaks down the stories and turn them into tasks.

### Daily Scrum

**(wikihow:Run a Daily Scrum)**The daily scrum is also known as the daily standup meeting. This serves to tighten communication and ensure that the entire team is on the same page. Each member goes through what they have done since the last standup, what they plan to work on before the next one, and outline any obstacles.

### Sprint retrospective meeting

Finally, after a sprint, the scrum master meets with the team for a retrospective meeting. They go over what went well, what did not, and what can be improved in the next sprint. The product owner is also present, and will listen to the team lay out the good and bad aspects of the sprint. This process allows the entire team to focus on its overall performance and identify strategies for improvement. It is crucial as the Scrum Master can observe common impediments and work to resolve them.

***(Advantage and disadvantage of Agile Scrum)***The advantage of SCRUM is compare to waterfall scrum can helps the company in saving time and money , because the development team does not need to wait too long to start development , Scrum also 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. It also have Fast moving, cutting edge developments can be quickly coded and tested using this method, as a mistake can be easily rectified. Scrum not only 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 and also like any other [agile methodology](http://www.my-project-management-expert.com/agile-methodology.html), which 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. The Daily meetings make it possible to measure individual productivity. This leads to the improvement in the productivity of each of the team members. The Issues are identified well in advance through the daily meetings and hence can be resolved in speedily and of course it is easier to deliver a quality product in a scheduled time. The most powerful thing is Scrum can work with any technology/ programming language but is particularly useful for fast moving web 2.0 or new media projects. And the most interesting part is the overhead cost in terms of process and management is minimal thus leading to a quicker, cheaper result.

One of disadvantage of SCRUM is, it 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. And 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. Thirdly if the team members are not committed, the project will either never complete or fail. And it only 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. And Scrum works well only 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 demoralization and the failure of the project. And also is if any of the team members leave during a development it can have a huge inverse effect on the project development. Lastly the [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.This methodology has use a lot because not only in game industry see this website below <http://stackoverflow.com/questions/1992194/do-any-of-you-use-scrum-for-game-development>

## Example of using SCRUM

1. We get all the request from the customers, executives , or even team members and write to the user stories

2. After collected all the user stories and put it into product backlog

3. Then the product owner we decide which user stories are going to put into the product

4. The Scrum master sets up meeting, monitors the work being done and facilitates release planning

5. The Team will start building the product building the sprint

6. Redo until finish

## Conclusion

|  |  |
| --- | --- |
| What is SCRUM | Scrum is an iterative and incremental agile software development framework for managing product development |
| Element of SCRUM | Product Backlog, Sprint Backlog, Burndown chart, Release Backlog   * Product Owner, Team, and [Scrum Master](http://scrummasterchecklist.org/pdf/ScrumMaster_Checklist_12_unbranded.pdf) * Sprint Planning, Daily Scrum, and Retrospective |
| 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. |
| Where has been used | Almost everywhere |

# Difference between agile and waterfall

## Content

Now we have a basic understanding of waterfall and agile let’s have a look at how a game industry work and compare between them and see which is more stible for the game industry.

|  |  |  |
| --- | --- | --- |
|  | Waterfall | 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

.

# Conclusion

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

# Reference

<http://www.onedesk.com/2013/06/scrum-methodology-vs-agile-methodology/>

**(Wikipedia:Waterfall model)*, Available from:***

<[***http://en.wikipedia.org/wiki/Waterfall\_model***](http://en.wikipedia.org/wiki/Waterfall_model)>

**Tutorialspoint. Available from:**

<<http://guides.is.uwa.edu.au/content.php?pid=43218&sid=318559>>

**Waterfall software and project development model, 2013 (video file), Available from:**

# <https://www.youtube.com/watch?v=\_W3R1m91Vdk>. [26 October 2012].

**NEW Intro to Agile Scrum in Under 10 Minutes - What is Scrum?), 2012(video file),Available from:**

<<https://www.youtube.com/watch?v=XU0llRltyFM>>

**(Wikipedia:Scrum(software development))** **Available from:** <<http://en.wikipedia.org/wiki/Scrum_(software_development)> >

**(Advantage and disadvantage of Agile Scrum), Available from:**

<<http://www.my-project-management-expert.com/the-advantages-and-disadvantages-of-agile-scrum-software-development.html>>

**(Wikipedia: System development life cycle)**

**<**<http://en.wikipedia.org/wiki/Systems_Development_Life_Cycle>>

**(wikihow:Run a Daily Scrum)**

**<**<http://www.wikihow.com/Run-a-Daily-Scrum>>