**Spiral Model**: Spiral Model is a Software development model that is more focused on interactive and risk approach rather than a document-driven or code-driven approach.

**Scenario:** I the author of this document and the owner of the companyoversees creating an application for the university registration system. We will have a team of 19 people including 12 software engineers. Our assignment for this project is to merge the BYUI grad plan with the registration system. We need to find an optimum schedule that will decide the number of sections needed for any specific class in that semester.

***Roles:***

* *Project manager*: The project manager will be Rochak: the owner of the company. To be the project manager one mush have a good management background and business-minded. Should be able to take hard and difficult decisions when needed. Rochak must do all the regular talking with the customer as well as present our product to our customers. Rochak should be good at managing his workers.
* *Custodian*: As regular custodian will be Chrissy. She will clean the office in the evening.
* *Secretaries:* Stan and Sally will be secretary and will help Rochak for office work. They will be very friendly and help to fill the bridge between Rochak and his worker.
* *Solution Engineer:* These people will be one of the most experienced in the whole team. He will come up with different solutions for customer requirements during the meeting which will be later explored by other engineers. Abe and Teri will be solution engineers.
* *Product Engineer*: This person will look for the product and try to work with people outside to see if the product they building will be liked by people. This person will also be very close to the customer.
* *Software engineer*: These people will plan the whole software. They will architect the implementation and working of the software. Larry, Britney, Claire be a software engineer. To get these roles one needs to have a good knowledge of software engineering as well as coding.
* *Developer and Tester*: Emily, Grace, Keith, Holly, Frank, and Dough are software developers. They will write code under the design and direction of software engineers. To be in this role one needs to be good at programming. Their job is to write code. They will also do testing and debugging.
* *Risk Analyst*: Ursula, Ingrid, and Jack will be a risk analyst. To be in this role one needs to know risk of doing business in different circumstances. They should also be fine in software engineering.

**Meetings:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Who | Agendas | Purpose | Event |
| 1st meeting / Life Cycle plan meeting | Rochak, Solution Engineer, Product Engineer | To assign everyone their role and try to find the optimum process for doing this project. To understand the project, to divide people into different groups. | To come up with different checkpoints and smalls goals that will guide us to our main goal without losing track. The checkpoint will be created for planning and documentation of the next phase, Risk analysis, prototyping, development, and testing, etc. We will try to get as much as information about the project if needed we will talk to Peter and Patricia | This overall project discussion meeting will be only once and will not be over until everyone is satisfied. |
| 2-times a week meeting | Everyone except custodian and secretary needs to attain this meeting. | Some changes that have come along or will come will be discussed. The technical or any other difficulty will be discussed. The checkpoint and the work done so far will be discussed. Any changes needed will be discussed. | To make sure everyone is satisfied with each other work. If some-one wants any changes in the project then they will be given a chance. The comments and opinions of the customer will be addressed. | Every Sunday and Wednesday at the start of the day. This meeting time is not exact and can go any long or short. |
| Daily team meeting | All the team members in a team will attain it. | To come up with the work plan for a day. | Every worker will have knowledge that what they are doing for the day. And at the end of the day what they should have in order to show for the next day's meeting. All team members need to discuss what they accomplished in last day of work. | This meeting will be held every day. |
| Requirement plan meeting. (This will be 2 phase meeting: info and requirement meeting) | Solution Engineer, Product Engineer, Software Engineer, Risk Engineer | To inform everyone about a new project.  To understand how to work with the customer and gather customer requirements and to identify objective. | Possible solutions to the requirements will be purposed to risk analyst after detail study on customer business and software requirements. | This meeting is held only once after everybody understands the projects and the team is formed. |
| Development plan meeting | Solution Engineer, Software Engineer, Developer, Risk Engineer | Understand the requirement validated in the previous phase. Come up with the general development idea. | To have some rough idea which path to follow for the development of software. Discuss the design and working of the software. Discuss the technology and tools to be used for software development. | To be held only once after the requirements are approved. |
| Integration and test plan meeting | Solution Engineer, Software Engineer, Developer, Risk Engineer | Understand the design validated in the previous phase. Come up with the general implementation idea. | Introduction of some strong way to implement the design. Discussion of some efficient way of the working process of software. Discussion about Detail designing, coding, testing, integration and testing, and review engineering. | To be held now and if customer unsatisfied start again from the requirement. |
| Risk analyst Meeting | Risk analyst | Evaluate the solution chosen by solution engineer and discussion on how to minimize risk for the chosen solution. | The best of the best solution is chosen and the possible risk is resolved and the final solution is sent to engineers to built prototype. | It happens every time after the requirement or development plan or integration plan passes through a solution engineer. |
| Programmer meeting | Software Engineer and Developer | Planning on software development and its testing | To discuss the idea to come up with bug-free partially working software. | It happens every time after the prototype is built. |
| Plan to fulfill customer’s unsatisfaction meeting | Solution Engineer, Product Engineer, Software Engineer, Risk Engineer | Understand the additional requirement and unsatisfaction from customers and redesign and reimplement. | Discussion about the changes asked by the customer. Understand the new requirement and discuss available solutions. | Jump to development plan meeting. But this time phase will be faster than the previous one. |

**Documentation:**Most of the time primary author will be Teri and Xavier. But all software engineers must also work on documentation and planning. All risk analysts will start working on purposing solutions with the lowest risk. The developer will start doing research about different development techniques under software engineer.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Authors | Audience | Purpose | Deadline and Usefulness |
| **Life Cycle Plan** | Main author- Rochak, Supporting author – Solution engineers, product engineer. | All workers | To understand the software and to organize the software development methodology and working pattern. | It should be completed before the requirement meeting. All the workers will able to understand how management has decided on the new project. |
| **Requirement plan** | Solution Engineers, Product Engineer, Software engineers | All engineers, risk analysts, and customers so they can ponder on what they really want. | To have a detailed understanding of customer business requirements and software requirements. | This should be completed before sending it to a solution engineer to look for best possible solutions for the requirements proposed. |
| **Development plan** | Teri, Product Engineer, Software Engineers | All engineer and risk analyst. | To have a detailed understanding of available options for developing our software. To have a detailed map of the working of software products. | This should be ready before sending it to a solution engineer to look for some good possible solutions for the development plan purposed. |
| **Integration and test plan** | Solution Engineers, Product Engineer, Software engineers | All engineer and risk analyst. | To have a blueprint of system integration. | This would be after the design is validated by the customer and before solution engineer would start looking for possible solutions in integrating full fledge software to the system. |
| **Requirement plan** | …………. | …………………… | ………………. | ……Repeat the same process but it would be a lot faster now………….. |

**Checkpoints:**

|  |  |  |
| --- | --- | --- |
| **Name** | **Time Estimation** | **Exit Criteria** |
| **Planning(It includes requirement plan, development plan, integration plan, next phase plan)** | Since the software is not big It should take around 1 week to finish it. And it is not expected to be perfect for the first time because we will visit it back. | All the business and software criteria of the customer are taken into account and the possible solution are purposed. |
| **Risk Analysis** | This part is heavily focused on this methodology so we will give this phase a week of duration. We will visit back this phase for every spiral. | After the solutions are evaluated and the best solution is chosen. The risk associated with that solution is resolved which is sent to an engineer for prototype development. |
| **Prototype** | We will spend half a week on this. | After the prototype is finished we can go to real development. |
| **Developing and Testing** | We will spend a month on this. But depends on how close we are to getting our software done. | Exit after the software is ready. |
| **Evaluation** | We will spend half a week on Evaluation. We will spend time making sure all the requirements are fulfilled before showing to our customer. | The evaluation must be done in every spiral. It is done by having a meeting with the customer. We take customer reviews and their new requirements if they have any. If they like the software then we are done and can exit. If not repeat the process form planning. |

The project is not big, and we have a small team; I personally don’t think the spiral method is the best technique for this project. We have a team of 19 and it is very easy to communicate in company so the simple software development model would have been more efficient. And also I feel like we will eventually waste a lot of time in making our software better and better which might not be worth the cost we will put in. The number of total phases and the spiral it can go through is unpredictable and for small companies like ours, we cannot afford such cost.

Reflection:   
 There are some strengths as well as weakness of the spiral model; some are listed below:

Strength

* Good for large projects.
* Flexibility in requirements
* Customer Satisfaction
* Software is produced early
* The best model for risk analysis and risk handling

Weakness

* Much more complex than any other software development mode.
* Not suitable for small projects
* It is expensive for small projects.
* Difficult in time management
* The number of phases and the number of spirals it can go through is unpredictable.

Resources:

* B. Boehm, "A Spiral Model of Software Development and Enhancement," *IEEE*, xxx, pp. 61–72, May 1988,   
  [Online] Available: [http://ieeexplore.ieee.org/xpls/abs\_all.jsp?arnumber=59&tag=1](https://content.byui.edu/file/fb36352f-44a4-473d-bb81-1e5a2ce36646/1/CS%20432%20PDFs/A%20Spiral%20Model%20of%20Software%20Dev.pdf)
* E. A. H. !!!, “Spiral Model in Hindi #7 || Software Engineering || MCS034 || BCS051,” *YouTube*, 03-Mar-2019. [Online]. Available: https://www.youtube.com/watch?v=A4s9lf46hM8. [Accessed: 04-Oct-2019].
* A. Powell-Morse, “Spiral Model: Software Development For Critical Projects,” *Airbrake Blog*, 02-Nov-2017. [Online]. Available: https://airbrake.io/blog/sdlc/spiral-model. [Accessed: 06-Oct-2019].
* E. 4u, “spiral model | software engineering |,” *YouTube*, 11-Jun-2018. [Online]. Available: https://www.youtube.com/watch?v=YfGvIhPXz1A&t=216s. [Accessed: 06-Oct-2019].

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Exceptional 100% | Good 90% | Acceptable 70% | Developing 50% | Missing 0% |
| Accuracy 40% | It is completely obvious which development methodology is described. Any knowledgeable person would be able to identify the methodology based on this document. | There is nothing to add and nothing wrong; the development methodology is completely described. One part of the plan may be misclassified as **bold** or *red/italic* | There exists one small problem (factual error or missing component). | There exists one large or multiple small problems (factual errors or missing components). | Large parts of the development methodology are inaccurately described or missing. |
| Application 30% | It is obvious that real thought went into the application (*the red/italic part*) of the plan. | The development methodology is applied to the scenario in an uncontrived way. | Every aspect of the scenario is incorporated into the development methodology. | Large parts of the plan are overly vague, do not appear to be related to the scenario, or do not appear to be related to the development methodology. | No attempt was made to apply the development methodology to the scenario. |
| Reflection 20% | The reflection cuts to the heart of the strengths and weaknesses of the development methodology. | The strengths and weakness of the development methodology are clearly communicated. | One strength and one weakness is mentioned in the reflection. | Little thought or effort was put in the reflection part of the paper. | The reflection part of the paper is missing. |
| Professionalism 10% | The paper is easy to read and ideas are clearly communicated. | Everything is properly cited, there are no grammar or spelling errors, and writing style is "professional." | One instance of a spelling error, grammar error, incomplete citation, overly verbose, poor formatting, or poor writing. | A citation is missing where one is needed (plagiarism alert!). |  |