

Agile Software Methodology

➤ Agile Methodology

- The agile software development model was proposed in the mid-1990s to overcome the serious shortcomings of the waterfall model of development identified above.
- The agile model was primarily designed to help a project to adapt to change requests quickly. Thus, a major aim of the agile models is to facilitate quick project completion.
- Agility is achieved by fitting the process to the project, i.e. removing activities that may not be necessary for a specific project. Also, anything that wastes time and effort is avoided.
- In the agile model, the requirements are decomposed into many small parts that can be incrementally developed. The agile model adopts an iterative approach. Each incremental part is developed over an iteration.

-
- At a time, only one increment is planned, developed, and then deployed at the customer site. No long-term plans are made. The time to complete an iteration is called a time box. The implication of the term time box is that the end date for an iteration does not change
 - Agile model emphasise face-to-face communication over written documents. It is recommended that the development team size be deliberately kept small (5–9 people) to help the team members meaningfully engage in face-to-face communication and have collaborative work environment. It is implicit then that the agile model is suited to the development of small projects.

PRESCRIPTIVE MODEL

- Also known as classical model , It is a set of predefined steps .
- Focuses on every activity equally.
- Takes more time that may not be realistic.
- Software configuration management is difficult.
- Results are produced with delay.

AGILE MODEL

- It is evolved over the period of time.
- Focuses on deliverables that fulfils user requirements.
- Takes much less time .
- Software configuration management is manageable.
- Quick results are produced.

-
- If a project being developed using waterfall model is cancelled mid-way during development, then there is nothing to show from the abandoned project beyond several documents.
 - Progress is generally measured in terms of the number of completed and reviewed artifacts such as requirement specifications, design documents, test plans, code reviews, etc. for which review is complete.

- Even if a project is cancelled midway, it still leaves the customer with some worthwhile code, that might possibly have already been put into live operation.
- Progress is measured in terms of the developed and delivered functionalities.

➤ **Agility Principles**

- Working software over comprehensive documentation.
- Frequent delivery of incremental versions of the software to the customer in intervals of few weeks.
- Requirement change requests from the customer are encouraged and are efficiently incorporated.
- Having competent team members and enhancing interactions among them is considered much more important than issues such as usage of sophisticated tools or strict adherence to a documented process. It is advocated that enhanced communication among the development team members can be realised through face-to-face communication rather than through exchange of formal documents.

-
- Continuous interaction with the customer is considered much more important rather than effective contract negotiation. A customer representatives is required to be a part of the development team, thus facilitating close, daily co-operation between customers and developers.

➤ Agile Testing Methodology

Early Testing focuses on :

- Writing test cases to express the behavior of the system .
- Early defect Prevention, Detection and removal.
- Ensuring that right test types are run at the right time.

Common Testing Methodologies are :

- Test driven development (TDD): Based on coding guided by tests.
- Acceptance test driven development (ATDD) : Based on communication between customers , developers , and testers and driven by predefined Acceptance criteria and Acceptance test cases.

-
- Behavior Driven Development (BDD) : Based on the expected behavior of the software being developed .

➤ Strengths

- The agile methods derive much of their agility by relying on the tacit knowledge of the team members about the development project and informal communications to clarify issues, rather than spending significant amounts of time in preparing formal documents and reviewing them.

➤ **Weakness**

- Lack of formal documents leaves scope for confusion and important decisions taken during different phases can be misinterpreted at later points of time by different team members.
- In the absence of any formal documents, it becomes difficult to get important project decisions such as design decisions to be reviewed by external experts.
- When the project completes and the developers disperse, maintenance can become a problem.

➤ **Importance of Agile methodology**

High product quality

In Agile methodology, testing is integrated throughout the cycle, which means that there are regular tests performed to see that the product is working during the development. This enables the product owner to make modification if needed and the developers are alert if there are any issues

Stakeholder Engagement

Agile presents multiple prospects for team engagement and stakeholder — after, before, and during each Sprint. By linking the client with each step of the project, there is an elevated amount of collaboration between the project team and client, providing more chances for the team to really understand the client's perspective. “Bringing working software near the beginning and commonly increases stakeholders' faith in the team's capability to deliver high-quality working software and supports them to be more deeply occupied in the project.”

Transparency

An exclusive opportunity for customers to be involved all through the project, from prioritizing characteristics to iteration planning and evaluation sessions to recurrent software builds enclosing new characteristics, these all are provided by agile development. However, this also involves clients to appreciate that they are seeing a job in progress in trade for this added benefit of lucidity.

➤ Agile Process Models

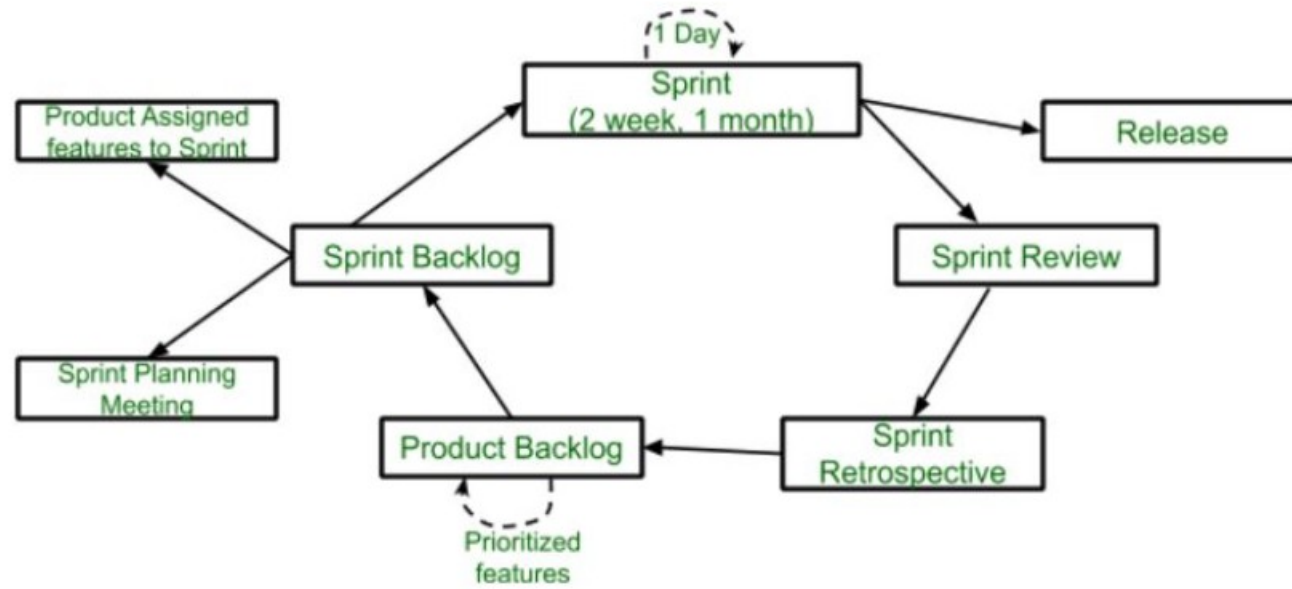
- Crystal
- Atern (formerly DSDM)
- Feature-driven development
- Scrum
- Extreme programming (XP)
- Lean development
- Unified process

➤ Scrum :

- Scrum advocates Whole Team approach , in the sense that every team member has to take part in every project activity.
- The whole team works together on Test Strategy, Test Planning, Test Specification, Test Execution, Test Evaluation, and Test Results Reporting.
- This approach also encourages proper use of the team talent instead of restricting to one activity. Testers also participate in all the project and development activities contributing their expertise in testing.
- In the scrum model, a project is divided into small parts of work that can be incrementally developed and delivered over time boxes that are called sprints.

-
- The software therefore gets developed over a series of manageable chunks. Each sprint typically takes only a couple of weeks to complete.
 - At the end of each sprint, stakeholders and team members meet to assess the progress made and the stakeholders suggest to the development team any changes needed to features that have already been developed and any overall improvements that they might feel necessary.
 - In the scrum model, the team members assume three fundamental roles— software owner, scrum master, and team member. The software owner is responsible for communicating the customers vision of the software to the development team. The scrum master acts as a liaison between the software owner and the team, thereby facilitating the development work.

➤ Scrum Process Flow



- **Sprint:**

A Sprint is a time-box of one month or less. A new Sprint starts immediately after the completion of the previous Sprint.

- **Release:**

When the product is completed then it goes to the Release stage.

- **Sprint Review:**

If the product still have some non-achievable features then it will be checked in this stage and then the product is passed to the Sprint Retrospective stage.

- **Sprint Retrospective:**

In this stage quality or status of the product is checked.

- **Product Backlog:**

According to the prioritize features the product is organized.

- **Sprint Backlog:**

Sprint Backlog is divided into two parts Product assigned features to sprint and Sprint planning meeting.

➤ Agile tools : IceScrum

- IceScrum is an online Scrum tool that offers Dashboard and Timeline views, Product Backlog, Release, and Sprint Plans, as well as Actors and Team functionality.
- It facilitates visual management with the help of virtual sticky notes.
- IceScrum is free for one team with an unlimited number of people, but the number of public projects is limited to just one in the free version.
- The software is a Java web application and requires Java container to run.

15 Project Team Menu: Dashboard Sandbox Product Backlog Timeline Members Sprint plan Vincent Barrier Logout

Project: IceScrum Role: ScrumMaster

Product Backlog

- 18 - Impossible de supprimer une éq... (1 pt)
- 149 - Manque suivant précédent dans ... (1 pt)
- 118 - Fichier avec IE8 (1 pt)
- 113 - comment bugs (1 pt)
- 53 - timeline avec sprints décalés (1 pt)

Release plan

Generate sprints Auto planning Dissociate all Vision Charts

sprint 4	sprint 5	sprint 6	sprint 7	sprint 8	sprint 9	sprint 10	sprint 11
0 / 2 14/2011	Sprint 7 - Done From 01/15/2011 to 02/04/2011	7 / 5	Sprint 8 - Done From 02/05/2011 to 02/18/2011	2 / 6	Sprint 9 - In progress From 02/19/2011 to 03/04/2011	0 / 4	Sprint 10 - Todo From 03/05/2011 to 03/18/2011
	<p>132 Bug diagramme flu... 1 Done</p> <p>48 Copier une story 1 Done</p> <p>12 Alertes par abonn... 1 Done</p> <p>185 Alert to start sp... 1 Done</p> <p>200 Date du jour sur ... 1 Done</p> <p>52 timeline pas cent...</p>		<p>117 Rôles excessifs 1 Done</p> <p>139 Règle métier sur ... 1 Done</p>		<p>8 Copy to Product B... 1 In progress</p> <p>151 End of sprint dat... 1 In progress</p> <p>186 Guide installatio... 1 In progress</p> <p>115 Plugin chat 1 In progress</p>		<p>116 Commentaires perd... 1 Planned</p> <p>30 End Of Sprint Rep... 1 Planned</p> <p>49 Publier toutes le... 1 Planned</p> <p>50 Impression plan d... 1 Planned</p> <p>148 Plus de couleurs ... 1 Planned</p> <p>138 Simplifié le chan...</p>

➤ DevOps

- DevOps is the combination of cultural philosophies, practices, and tools that increases an organization's ability to deliver applications and services at high velocity: evolving and improving products at a faster pace than organizations using traditional software development and infrastructure management processes.
- This speed enables organizations to better serve their customers and compete more effectively in the market.
- DevOps is a new term emerging from the collision of two major related trends. The first was also called “agile infrastructure” or “agile operations”; it sprang from applying Agile and Lean approaches to operations work. The second is a much expanded understanding of the value of collaboration between development and operations staff throughout all stages of the development

lifecycle when creating and operating a service, and how important operations has become in our increasingly service-oriented world .

- DevOps is a set of practices that combines software development (*Dev*) and IT operations (*Ops*). It aims to shorten the system development life cycle and provide continuous delivery with high software quality.
- Agile and DevOps serve complementary roles: several standard DevOps practices such as automated build and test, continuous integration, and continuous delivery originated in the Agile .
- Agile can be viewed as addressing communication gaps between customers and developers, while DevOps addresses gaps between developers and IT operations / infrastructure. Also, DevOps has focus on the deployment of developed software, whether it is developed via Agile or

Waterfall model	Incremental model	Prototyping model	V model
<ul style="list-style-type: none"> • It takes more time than all other models 	<ul style="list-style-type: none"> • Takes less time than waterfall model 	<ul style="list-style-type: none"> • Takes less time than incremental model 	<ul style="list-style-type: none"> • Takes less time than waterfall model
<ul style="list-style-type: none"> • Linear approach 	<ul style="list-style-type: none"> • Iterative approach 	<ul style="list-style-type: none"> • Iterative approach 	<ul style="list-style-type: none"> • Linear approach
<ul style="list-style-type: none"> • Requirements must be known prior • classical waterfall model cannot be used in practical development projects 	<ul style="list-style-type: none"> • Needs good planning and design 	<ul style="list-style-type: none"> • Difficult to divide in parts • The evolutionary approach is suitable for large problems which can be decomposed into a set of modules for incremental development and delivery. This model is also used widely for object-oriented development projects 	<ul style="list-style-type: none"> • Requirements must be known prior

Extreme Programming Model (XP)

- This model is based on a rather simple philosophy: "If something is known to be beneficial, why not put it to constant use?" Based on this principle, it puts forward several key practices that need to be practiced to the extreme.
- XP is based on frequent releases (called iteration), during which the developers implement "user stories". User stories are similar to use cases, but are more informal and are simpler. A user story is the conversational description by the user about a feature of the required.
- For example, a user story about a library software can be:
 - A library member can issue a book.
 - A library member can query about the availability of a book.
 - A library member should be able to return a borrowed book.

➤ **XP Activities :**

- Coding
- Testing
- Listening
- Designing
- Feedback
- Simplicity

adaptive software development

Adaptive Software Development is a method to build complex software and system. ASD focuses on human collaboration and self-organization. ASD “**life cycle**” incorporates three phases namely:

Speculation:

During this phase project is initiated and planning is conducted. The project plan uses project initiation information like project requirements, user needs, customer mission statement etc, to define set of release cycles that the project wants.

Collaboration:

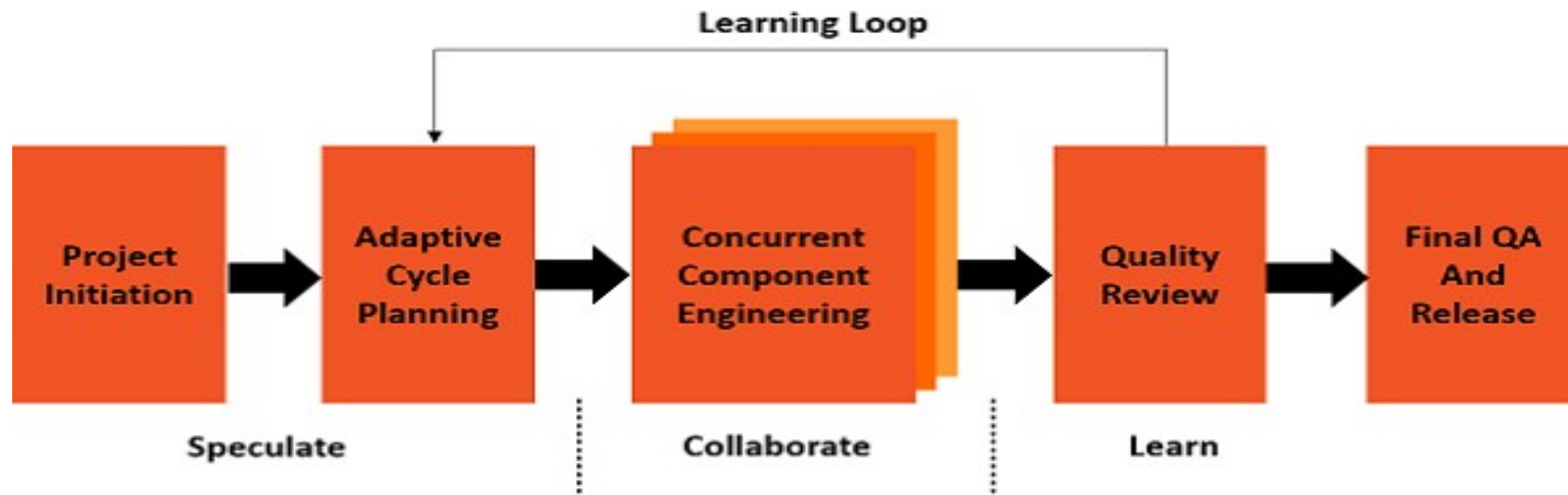
It is the difficult part of ASD as it needs the workers to be motivated. It collaborates communication and teamwork but emphasizes individualism as individual creativity plays a major role in creative thinking. People working together must trust each others to.

Learning:

The workers may have a overestimate of their own understanding of the technology which may not lead to the desired result. Learning helps the workers to increase their level of understanding over the project.

Learning process is of 3 ways:

ASD

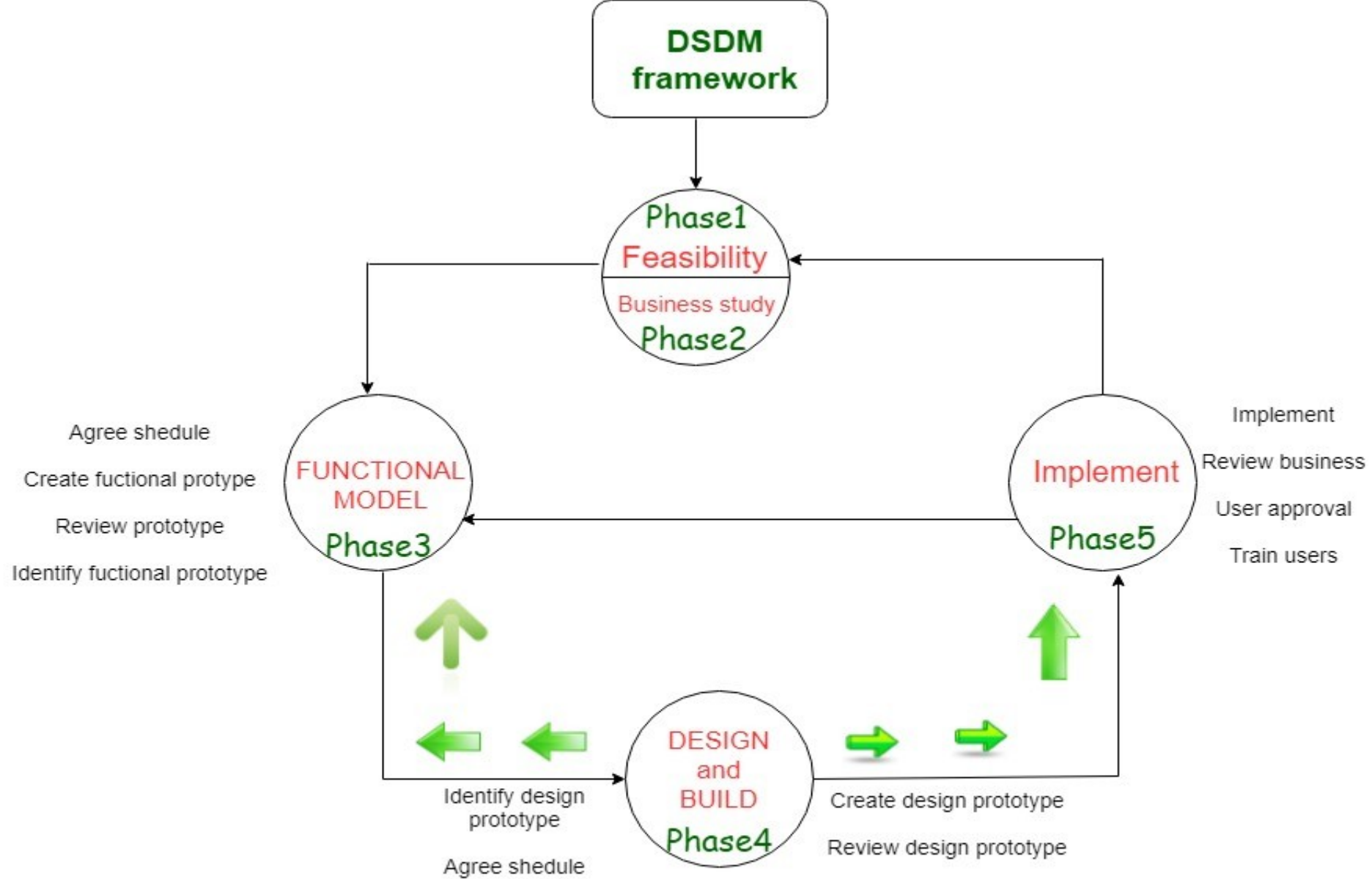


dynamic system development method

The **Dynamic Systems Development technique (DSDM)** is an associate degree agile code development approach that provides a framework for building and maintaining systems.

DSDM is An iterative code method within which every iteration follows the 80% rule that simply enough work is needed for every increment to facilitate movement to the following increment. The remaining detail is often completed later once a lot of business necessities are noted or changes are requested and accommodated.

the DSDM life cycle that defines 3 different unvarying cycles, preceded by 2 further life cycle activities:



Dynamic Systems Development Method life cycle

Feasibility Study:

It establishes the essential business necessities and constraints related to the applying to be designed then assesses whether or not the application could be a viable candidate for the DSDM method.

Business Study:

It establishes the use and knowledge necessities that may permit the applying to supply business value; additionally, it is the essential application design and identifies the maintainability necessities for the applying.

Functional Model Iteration:

It produces a collection of progressive prototypes that demonstrate practicality for the client.

Design and Build Iteration:

It revisits prototypes designed throughout useful model iteration to make sure that everyone has been designed during a manner that may alter it to supply operational business price for finish users. In some cases, useful model iteration and style and build iteration occur at the same time.

Implementation:

It places the newest code increment (an “operationalized” prototype) into the operational surroundings. It ought to be noted that:

- **(a)** the increment might not 100% complete or,
- **(b)** changes are also requested because the increment is placed into place. In either case, DSDM development work continues by returning to the useful model iteration activity.

JIRA SOFTWARE

JIRA is a software testing tool developed by the Australian Company Atlassian.

It is a bug tracking tool that reports all the issues related to your software or mobile apps. The word JIRA comes from the Japanese word, i.e., "Gojira" which means Godzilla.

Projects: It is used to manage the defects very effectively.

Issue: It is used to track and manage the defects/issues.

Aspects



Workflow: Processes the Issue/Defect life cycle. Suppose we have a business requirement, we create the technical design and from the technical design, we create the test cases. After creating the test cases, coding is done, and then testing is performed on the project. This design workflow is possible by using Jira.

Search: Find with ease. Suppose we have done with a project at the beginning of the December and its version is 1.0. Now, we move to version 1.1 and completed at the end of December. What we are doing is that we are adding new versions. Through Jira, we can get to know that what happened in the earlier versions, how many defects occurred in the earlier projects and the learning we achieve from the earlier projects.

Dashboards: Dashboard is a display which you see when you log in to the Jira. You can create multiple dashboards for multiple projects. You can create the personal dashboard and can add the gadgets in a dashboard so that you can keep track of the assignments and issues that you are working on.

Why we used JIRA

Plan, Track and Work Faster

JIRA is a bug-tracking tool mainly used to track, organize, and prioritize the bugs, newly added features, improvements for certain software releases.

The main source of information

JIRA is the primary source of information for the next software release. On JIRA, the whole team of the software developers can plan for the new features which are to be added and bugs to be fixed in the next release.

Organize the documentation tasks

JIRA tool is used to organize the documentation tasks. It is useful in grouping the multiple tasks by using the component functionality, and even you can create your own documentation. In this way, you can create a structured way of documentation.

Track the progress of our documentation

It is a very useful tool in tracking the progress of our documentation. JIRA tool provides a very important feature, i.e., pie chart macro. In the pie chart macro, you can view tasks such as Open tasks, Closed tasks, Resolved tasks.

Track the progress of our documentation

It is a very useful tool in tracking the progress of our documentation. JIRA tool provides a very important feature, i.e., pie chart macro. In the pie chart macro, you can view tasks such as Open tasks, Closed tasks, Resolved tasks.

Measures the time spent on documentation

JIRA tool does not have the default functionality for measuring the time spent on documentation. JIRA tool is bundled with the Tempo Timesheets, which measures how much time has been spent on the documentation.

Provides feedback faster

JIRA tool provides the Confluence pages where you can connect to the issues in just a few clicks. If something needs to be updated, then you can create the issues directly from the Confluence page.