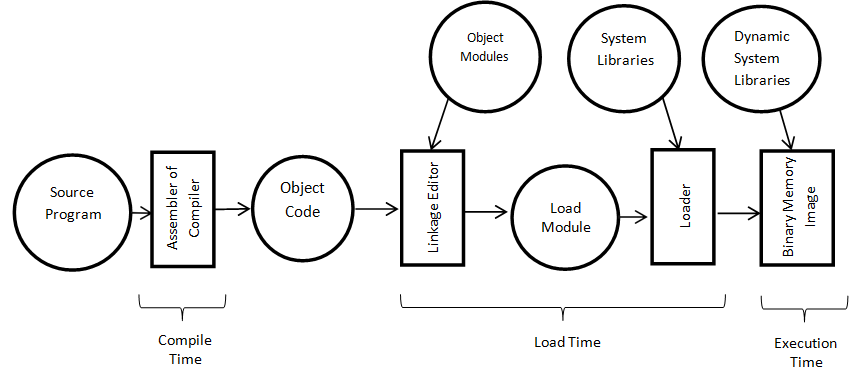
#### **Life Cycle of Source Program**



The life cycle of a source program defines the program behavior and extends through execution stage, which exhibits the behavior specified in the program.

Every source program goes through a life cycle of several stages.

* **Edit time:** It is the phase where editing of the program code takes place and is also known as design time. At this stage, the code is in its raw form and may not be in a consistent state.
* **Compile time:** At the compile time stage, the source code after editing is passed to a translator that translates it into machine code. One such translator is a compiler. This stage checks the program for inconsistencies and errors and produces an executable file.
* **Distribution time:** It is the stage that sends or distributes the program from the entity creating it to an entity invoking it. Mostly executable files are distributed.
* **Installation time:** Typically, a program goes through the installation process, which makes it ready for execution within the system. The installation can also optionally generate calls to other stages of a program's life cycle.
* **Link time:** At this stage, the specific implementation of the interface is linked and associated to the program invoking it. System libraries are linked by using the lookup of the name and the interface of the library needed during compile time or throughout the installation time, or invoked with the start or even during the execution process.
* **Load time:**This stage actively takes the executable image from its stored repositories and places them into active memory to initiate the execution. Load time activities are influenced by the underlying operating system.
* **Run time:** This is the final stage of the life cycle in which the programmed behavior of the source program is demonstrated.

#### **Source Code Management Tools**

There are different source code management tools available for use and a lot of them are also open-source

and free to use. Depending on the team’s requirements, an appropriate tool can be used.

Tools integrating project management and build pipeline features are GitLab and Team Foundation Server.

For on-premise setup, teams can use Git, CVS, etc.

#### List Of The Best Source Code Management Tools

**Here is the list of popular source code versioning and management tools:**

1. GitHub
2. Git
3. GitLab
4. Apache Subversion (SVN)
5. CVS
6. Mercurial
7. Monotone
8. Bitbucket Server
9. Team Foundation Server (TFS)
10. Bazaar

#### Comparing Top Source Code Management Software

|  |  |  |
| --- | --- | --- |
| **Tool** | **Features** | **Pricing** |
| [**Github**](https://github.com/) | Enables large development teams to collaborate, review, and manage software or application code | • Offers free trial  • Team: $4 per user/month  • Enterprise: $21 per user/month |
| [**Bitbucket**](https://bitbucket.org/) | One-stop solution for versioning, project management, and collaboration across teams of any size | • Free for small teams with up to 5 members  • Standard and Premium come at $3 and $6 per user/month, respectively |
| [**Gitlab**](https://about.gitlab.com/) | Used for end-to-end project lifecycle with git-based tools: version control, project management, CI/CD | • Free for individuals  • Premium and Ultimate editions come at a cost of $19 and $99 per user/month, respectively |
| [**Team Foundation Server**](https://azure.microsoft.com/en-us/services/devops/server/) | Enterprise-grade source control management tool that supports integration with most of the existing IDEs | • Offers free trial  • Basic Plan: $2 per user/month  • Azure Pipelines: $15 per parallel job |
| [**Apache Subversion**](https://subversion.apache.org/) | Open-source version control system supporting file locking and merge tracking | • Open source: deployed on premise and is free to use |

we learned about different source code management tools, their features, and their pros and cons.

Version control systems are one of the most important tools used by almost all the development teams to keep a track of changes to the application code and help multiple team members to collaborate on the same project simultaneously.

The most commonly based and adopted tools by teams of small and large enterprises are **GitHub** and **GitLab** which provide almost all the required features under one roof.

Both can provide secure cloud-based hosting solutions for repositories and great integration with CI/CD tools and defect management tools. They also provide an integrated mechanism for Pull requests and reviews.

All these features make these a preferred choice for most of the new projects and organizations.

References

1. [Life Cycle of Source Program (information-gtu.blogspot.com)](https://information-gtu.blogspot.com/2020/05/life-cycle-of-source-program.html)[What Is The Project Life Cycle: The 5 Phases & Why It’s Important (thedigitalprojectmanager.com)](https://thedigitalprojectmanager.com/project-management-life-cycle/)
2. [10 Best Source Code Management Tools For Version Control (softwaretestinghelp.com)](https://www.softwaretestinghelp.com/best-source-code-management-tools/#:~:text=Answer%3A%20There%20are%20different%20source%20code%20management%20tools,pipeline%20features%20are%20GitLab%20and%20Team%20Foundation%20Server.)