

UNIT - III

Software Construction

Disclaimer:

The lecture notes have been prepared by referring to many books and notes prepared by the teachers. This document does not claim any originality and cannot be used as a substitute for prescribed textbooks.

Topics

- Software Construction
- Coding Standards
- Coding Framework
- Reviews – Desk Checks (Peer Reviews)
- Walkthroughs
- Code Reviews, Inspections
- Coding Methods
- Structured Programming
- Object-Oriented Programming
- Automatic Code Generation
- Software Code Reuse
- Pair Programming
- Test-Driven Development
- Configuration Management
- Software Construction Artefacts

Software Construction

Introduction

- A layman believes that software construction is the entire software development process.
- But, in fact, it is just one of the crucial tasks in software development; software requirement management, software design, software testing and software deployment are all equally crucial tasks.
- Furthermore, the process of software construction itself consists of many tasks; it not only includes software coding but also unit testing, integration testing, reviews and analysis.
- Construction is one of the most labor intensive phases in the software development life cycle.
- It comprises 30% or more of the total effort in software development.
- What a user sees as the product at the end of the software development life cycle is merely the result of the software code that was written during software construction.
- Due to the labor intensive nature of the software construction phase, the work is divided not only among developers, but also small teams are formed to work on parts of the software build.

Software Construction

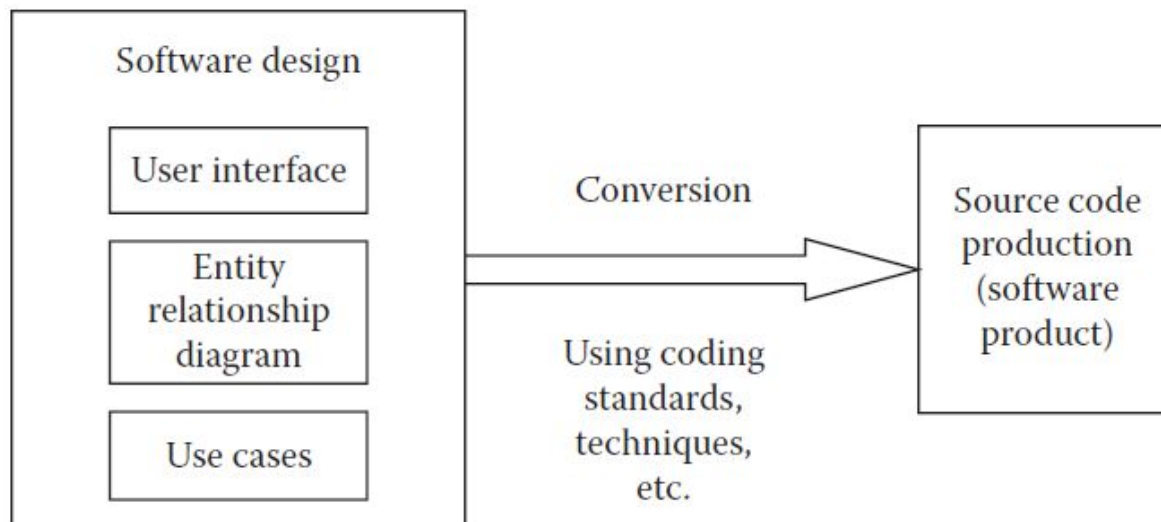
Introduction

- In fact, to shrink the construction time, many distributed teams, either internal or through contractors are deployed.
- The advantage to this is that these project teams do the software coding and other construction work in parallel with each other and thus the construction phase can be collapsed.
- This parallel development is known as concurrent engineering.
- Constructing an industry strength software product of a large size requires stringent coding standards.
- The whole process of construction should follow a proven process so that the produced code is maintainable, testable and reliable.
- The process itself should be efficient so that resource utilization can be optimized and thus cost of construction can be kept at a minimum.

Software Construction

Coding Standards

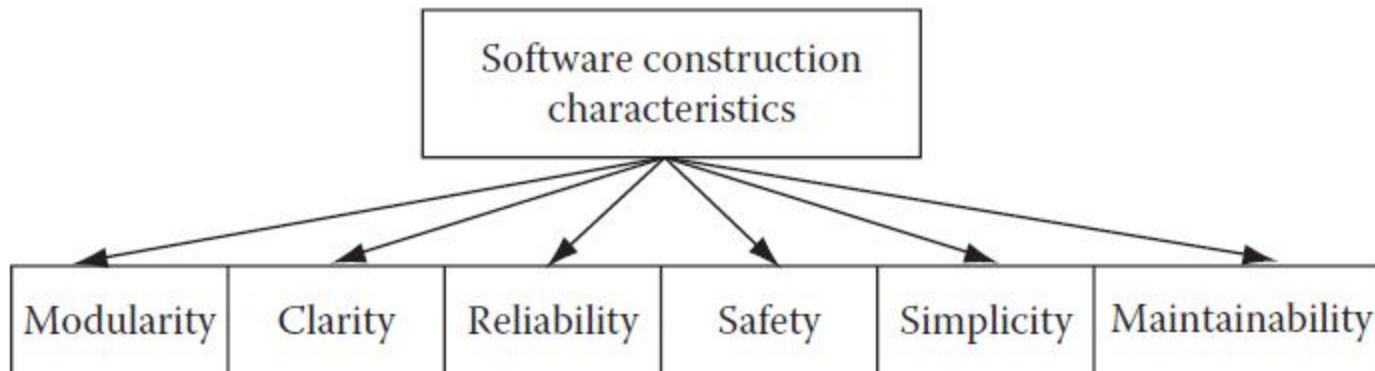
- Developers are given software design specifications in the form of use cases, flow diagrams, UI mock ups, etc., and they are supposed to write a code so that the built software matches these specifications.
- Converting the specifications into software code is totally dependent on the construction team.
- How well they do it depends on their experience, skills and the process they follow to do their job.
- Apart from these facilities, they also need some standards in their coding so that the work is fast as well as has other benefits like maintainability, readability and reusability (Figure-Source Code Production (Conversion) from Software Design).



Software Construction

Coding Standards

- At any time, a code written by a developer will always be different from that written by any other developer.
- This poses a challenge in terms of comprehending the code while reusing the code, maintaining it, or simply reviewing it.
- A uniform coding standard across all construction teams working on the same project will make sure that these issues can be minimized if not eliminated (Figure below - Software Construction Characteristics).
- Some of the coding standards include standards for code modularity, clarity, simplicity, reliability, safety and maintainability.



Software Construction

Coding Standards – Modularity

- The produced software code should be modular in nature.
- Each major function should be contained inside a software code module.
- The module should contain not only structure, but it should also process data.
- Each time a particular functionality is needed in the software construction, it can be implemented using that particular module of software code.
- This increases software code reuse and thus enhances productivity of developers and code readability.

Coding Standards – Clarity

- The produced code should be clear for any person who would read the source code.
- Standard naming conventions should be used so that the code has ample clarity.
- There should be sufficient documentation inside the code block, so that anybody reading the code could understand what a piece of code is supposed to do.
- There should also be ample white spaces in the code blocks, so that no piece of code should look crammed. White spaces enhance readability of written code.

Software Construction

Coding Standards – Simplicity

- The source code should have simplicity and no unnecessary complex logic; improvisation should be involved, if the same functionality can be achieved by a simpler piece of source code.
- Simplicity makes the code readable and will help in removing any defects found in the source code.
- Simplicity of written code can be enhanced by adopting best practices for many programming paradigms.
- For instance, in the case of object-oriented programming, abstraction and information hiding add a great degree of simplicity.
- Similarly, breaking the product to be developed into meaningful pieces that mimic real life parts makes the software product simple.

Coding Standards – Reliability

- Reliability is one of the most important aspects of industry strength software products.
- If the software product is not reliable and contains critical defects, then it will not be of much use for end users.

Software Construction

Coding Standards – Reliability

- Reliability of source code can be increased by sticking to the standard processes for software construction.
- During reviews, if any defects are found, they can be fixed easily if the source code is neat, simple, and clear.
- Reliable source code can be achieved by first designing the software product with future enhancement in consideration as well as by having a solid structure on which the software product is to be built.
- When writing pieces of source code based on this structure, there will be little chance of defects entering into the source code.
- Generally during enhancements, the existing structure is not able to take load of additional source code and thus the structure becomes shaky.
- If the development team feels that this is the case, then it is far better to restructure the software design and then write a code based on the new structure than to add a spaghetti code on top of a crumbling structure.

Software Construction

Coding Standards – Safety

- Safety is important, considering that software products are used by many industries where human lives are concerned and that human lives could be in danger because of faulty machine operation or exposure to a harmful environment.
- In these industries, the software product must be ensured to operate correctly and chances of error are less than 0.00001%.
- Industries like medicine and healthcare, road safety, hazardous material handling need foolproof software products to ensure that either human lives are saved (in case of medicine and healthcare) or human lives are not in danger.
- Here the software code must have inbuilt safety harnesses.

Coding Standards – Maintainability

- As it has been pointed out after several studies, maintenance costs are more than 70% of all costs including software development, implementation, and maintenance.
- To make sure that maintenance costs are under limit during software construction, it should be made sure that the source code is maintainable.
- It will be easy to change the source code for fixing defects during maintenance.

Software Construction

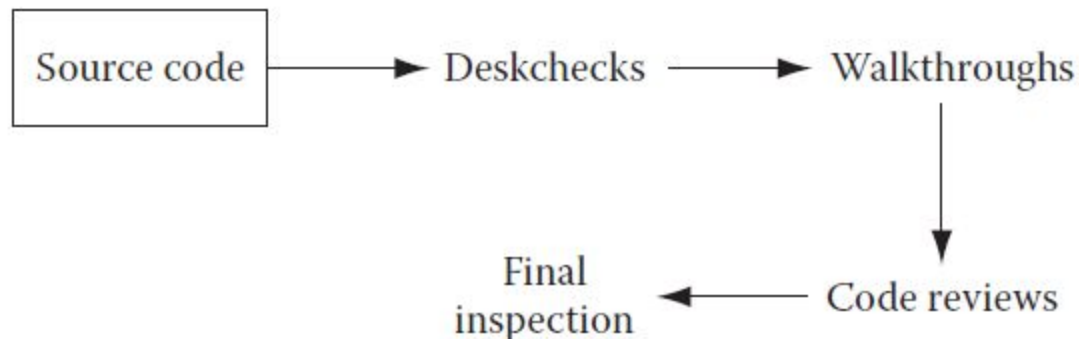
Coding Framework

- Like most construction work, you need to set up an infrastructure based on which construction can take place.
- For software construction, you need to have a coding framework that will ensure a consistent coding production with standard code that will be easy to debug and test.
- In object oriented programming, what base classes are to be made, which will be used throughout construction, is a subject that is part of the coding framework.
- In general, coding frameworks allow construction of the common infrastructure of basic functionality which can be extended later by the developers.
- This way of working increases productivity and allows for a robust and well structured software product.
- It is similar in approach to house building where a structure is built based on a solid foundation.

Software Construction

Reviews (Quality Control)

- It is estimated that almost 70% of software defects arise from faulty software code.
- To compound this problem, software construction is the most labor intensive phase in software development.
- Any construction rework means wasting a lot of effort already put in.
- Moreover, it is also a fact that it is cheaper to fix any defects found during construction at the phase level itself.
- If those defects are allowed to go in software testing (which is the next phase), then fixing those defects will become costlier.
- That is why review of the software code and fixing defects is very important.
- There are some techniques available like deskchecks, walkthroughs, code reviews, inspections, etc. that ensure quality of the written code (Figure below-Source code review methods and their operation sequence).



Software Construction

Reviews (Quality Control)

- These different kinds of reviews are done at different stages in software code writing.
- They also serve different purposes.
- While inspections provide the final go/no go decision for approval of a piece of code, other methods are less formal and are meant for removing defects instead of deciding whether a piece of code is good enough or not.

Reviews – Deskchecks (Peer Reviews)

- Deskchecks are employed when a complete review of the source code is not important.
- Here, the developer sends his piece of code to the designated team members.
- These team members review the code and send feedback and comments to the developer as suggestions for improvement in the code.
- The developer reads those feedbacks and may decide to incorporate or to discard those suggestions.
- So this form of review is totally voluntary.
- Still, it is a powerful tool to eliminate defects or improve software code.

Software Construction

Reviews – Walkthroughs

- Walkthroughs are formal code reviews initiated by the developer. The developer sends an invitation for walkthrough to team members.
- At the meeting, the developer presents his method of coding and walks through his piece of code.
- The team members then make suggestions for improvement, if any.
- The developer then can decide to incorporate those suggestions or discard them.

Reviews – Code Reviews

- Code reviews are one of the most formal methods of reviews. The project manager calls for a meeting for code review of a developer.
- At the meeting, team members review the code and point out any code errors, defects, or improper code logic for likely defects. An error log is also generated and is reviewed by the entire team.

Reviews – Inspections

- Code inspections are final reviews of software code in which it is decided whether to pass a piece of code for inclusion into the main software build.

Software Construction

Coding Methods

- Converting design into optimal software construction is a very serious topic that has generated tremendous interest over the years.
- Many programming and coding methods were devised and evolved as a result.
- As it is well known in the industry, the early software products were of small size due to limited hardware capacity.
- With increasing hardware capacity, the size of software products has been increasing.
- Software product size affects the methods that can be used to construct specific sized software products.
- Advancement in the field of computer science also allows discovery of better construction methods.
- To address needs of different sized software products in tandem with advancement in computer science, different programming techniques evolved.
- These include structured programming, object-oriented programming, automatic code generation, test-driven development, pair programming, etc.

Software Construction

Coding Methods – Structured Programming

- Structured programming evolved after mainframe computers became popular.
- Mainframe computers offered vast availability of computing power compared to primitive computers that existed before.
- Using structured programming, large programs could be constructed that could be used for making large commercial and business applications.
- Structured programming enabled programmers to store large pieces of code inside procedures and functions.
- These pieces of code could be called by any other procedures or functions.
- This enabled programmers to structure their code in an efficient way.
- Code stored inside procedures could be reused anywhere in the application by calling it.

Software Construction

Coding Methods – Object-Oriented Programming

- In structured programming, data and structured code are separate and accordingly they are modeled separately.
- This is an unnatural way of converting real life objects into software code because objects contain both data and structure.
- Widely used as an example in object-oriented programming books, a car consists of a chassis, an engine, four wheels, body, and transmission.
- Each of these objects has some specific properties and specific functions.
- When a software system is modeled to represent real-world objects, both data and structure are taken care of in object-oriented programming.
- From outside of a class that is made to represent an object, only the behavior of the object is visible or perceived.
- Unnecessary details about the object are hidden and in fact are not available from outside.
- This kind of representation of objects makes them robust and a system built on using them has relatively few problems.

Software Construction

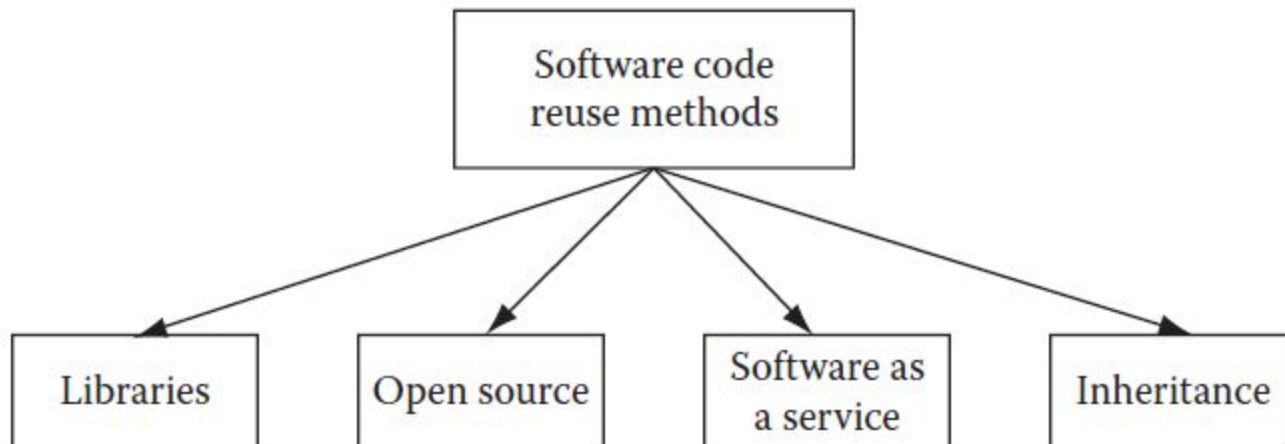
Coding Methods – Automatic Code Generation

- Constructing and generating software code is very labor intensive work. So there has always been fascination about automatic generation of software code.
- Unfortunately, this is still a dream. Some CASE(computer Assisted Self Evaluation) and modeling tools are available that generate software code. But they are not sophisticated. They are also not complete.
- Then there are business analyst platforms developed by many ERP software vendors that generate code automatically when analysts configure the product.
- These analyst platforms are first built using any of the software product development methodologies.
- The generated code is specific to the platform and runs on the device (hardware and software environment) for which the code is generated.
- Generally, any code consists of many construction unit types.
- Some of these code types include control statements such as loop statements, if statements, etc., and database access, etc.
- Generating all of the software code required to build a software application is still difficult.
- But some companies like Sun Microsystems are working to develop such a system.

Software Construction

Coding Methods – Software Code Reuse

- Many techniques have evolved to reduce the labor intensive nature of writing source code.
- Software code reuse is one such technique.
- Making a block of source code to create a functionality or general utility library and using it at all places in the source code wherever this kind of functionality or utility is required is an example of code reuse.
- Code reuse in procedural programming techniques is achieved by creating special functions and utility libraries then using them in the source code.
- In object-oriented programming, code reuse is done at a more advanced level.



Software Construction

Coding Methods – Software Code Reuse

- The classes containing functions and data themselves can not only be reused in the same way as functions and libraries but the classes can also be modified by way of creating child classes and using them in the source code (Figure above – Code reuse methods).
- Apart from creating and using libraries and general purpose classes for code reuse, a more potent code reuse source has evolved recently.
- It is known as “service oriented architecture” (SOA).

Software Construction

Coding Methods – Pair Programming

- Pair programming is a quality driven development technique employed in the eXtreme Programming development model.
- Here, each development task is assigned to two developers.
- While one developer writes the code, the other developer sits behind him and guides him through the requirements (functional, nonfunctional).
- When it is the turn of the other developer to write the code, the first developer sits behind him and guides him on the requirements.
- So developers take turns for the coding and coaching work.
- This makes sure that each developer understands the big picture and helps them to write better code with lesser defects.

Software Construction

Coding Methods – Test-Driven Development

- This concept is used with iteration-based projects especially with eXtreme Programming technique.
- Before developers start writing source code, they create test cases and run the tests to see if they run properly and their logic is working.
- Once it is proved that their logic is perfect, only then they write the source code.
- So here, tests drive software development, and hence it is appropriately named test-driven development.

Software Construction

Configuration Management

- Configuration management plays an important role in the construction phase.
- Due to changes in requirements and design, an already developed source code needs to be changed.
- So it happens that the development team ends up with many versions of a source code during the project.
- If the version control management is not handled properly, then many developers may start working on a wrong version of source code, and thus a lot of rework may be needed in the end.
- There is one more dimension to configuration management for the construction phase.
- During construction, many software builds are maintained for different versions of the product being developed.
- These builds can break if a bad piece of code is checked into the build by any developer.
- When the build is broken, then no other developer can check in his code.
- Thus, development is halted until the build is rebuilt with the correct code.

Software Construction

Configuration Management

- Imagine what may happen in the case of distributed teams located at far-flung locations with different time zones and a central build is being maintained.
- It will be difficult to communicate and manage the build process in such a scenario.
- In such scenarios, smoke test application can be deployed, which can run whenever a new code is checked-in in the build.
- If the smoke test fails, that means the build has failed and thus the automated system can e-mail the build information to concerned people.
- If the build fails, then the developer who had checked-in in the code gets the message and immediately tries to fix the build.
- Once the build is fixed, then other developers can check-in their code.
- Thus, configuration management plays an important role in construction phase.

Software Construction

Software Construction Artifacts

- The software construction phase is one of the most labor intensive phases in software development cycle.
- This phase generates the complete source code of the application.
- Apart from source code, documentation is also made so that when any maintenance is required on the built application, the source code could be well understood, and changing any source code will be easy.
- Review reports are also generated after reviews are conducted.

REFERENCES

- Ashfaque Ahmed, Software Project Management: A Process-driven approach, Boca Raton, Fla: CRC Press, 2012.

THANK YOU