SE 323 - Software Construction Testing and Maintenance

MODULE ONE
SOFTWARE CONSTRUCTION

CHAPTER ONE
OVERVIEW

Module 1
Software Construction

OUTLINE

1. *Overview of Software Construction*
2. *Planning/Preparing for Software Construction*
3. *Doing Software Construction*
4. *Documenting Software Construction*
5. *Measuring/Monitoring Software Construction*

Module 1
Software Construction

OUTLINE

1. *Overview of Software Construction*
2. ~~*Planning/Preparing for Software Construction*~~
3. ~~*Doing Software Construction*~~
4. ~~*Documenting Software Construction*~~
5. ~~*Measuring/Monitoring Software Construction*~~

Software Construction
OVERVIEW

OUTLINE

1. *What is software construction?*
2. *Why is software construction important?*
3. *How to do software construction?*

Software Construction
OVERVIEW

OUTLINE

1. *What is software construction?*
2. ~~*Why is software construction important?*~~
3. ~~*How to do software construction?*~~

Software Construction – What?

What is Software Construction?

- By Definition
- By Metaphors
- By Context

Software Construction – What?

What is Software Construction?

- By Definition
- ~~By Metaphors~~
- ~~By Context~~

Software Construction - Definition

DEFINITION: literal meaning

To "construct" means:

- To make
- To build
- To pile up together
- To put or fit something together
- To draw a line, figures, etc in agreement with certain rules

---

*Ref: *"The American Heritage Dictionary of the English Language"*

"Oxford Advanced Learner's Dictionary of Current English"

Software Construction – What?

What is Software Construction?

- ~~By Definition~~
- By Metaphors
- ~~By Context~~

Software Construction - Metaphors

Metaphors

- A modeling tool
- An analogy used to describe something less familiar by comparing it with something already familiar
- Heuristic vs. Algorithm
  - e.g. searchlight vs. roadmap
- Not mutually exclusive
  - Can be used in combination
- Structural framework for human behaviors

Software Construction - Metaphors

Writing Metaphor

Software Development = "Writing Code"

Software Penmanship: Writing Code

- A casual letter: writing from start to finish, no formal planning
- Literate program: novel writing
- One-person activity (many in software)
- Fully complete, difficult to change (2/3 in software)
- High premium on originality (reuse in software)
- Planning to throw one away
- *Too simple and rigid as a software development process*

Software Construction - Metaphors

Farming Metaphor

Software Development = "Growing, Farming"

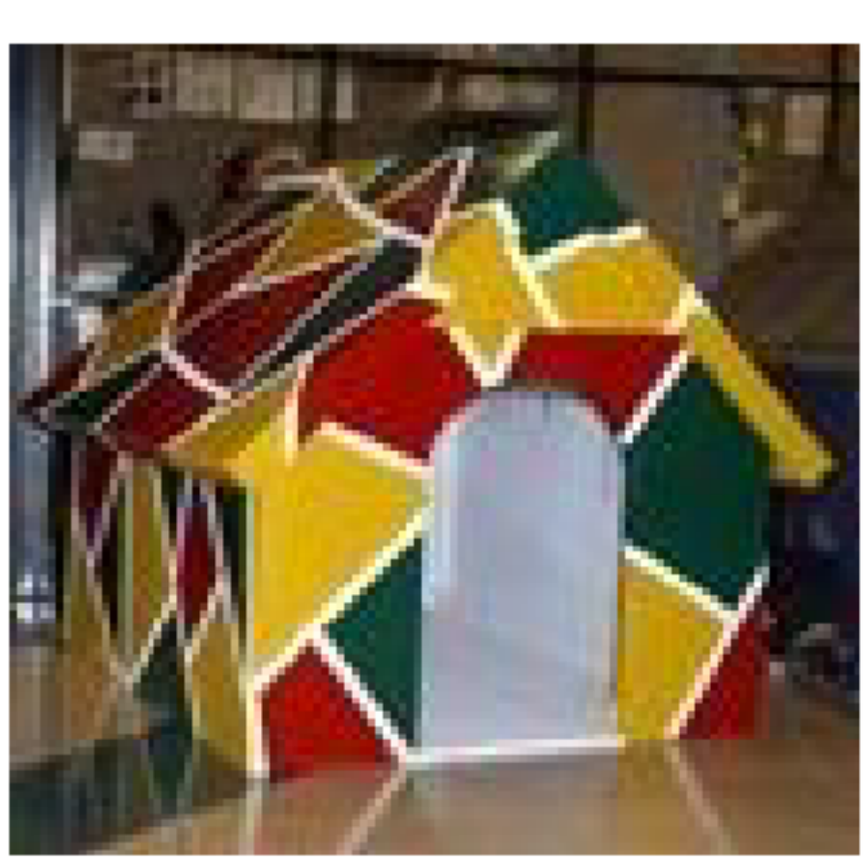
- **Software Farming: Growing a system**
  - Planting a seed, growing a crop
  - Incremental technique - adding "little steps" to the system
  - No direct control over how the software develops
- **Software Oyster Farming: System Accretion**
  - Increase in size by a gradual external addition or inclusion
  - Incremental, iterative, adaptive and evolutionary
  - Forming a skeleton - like the oyster's beginning a pearl
  - Doesn't overpromise – good incremental development model

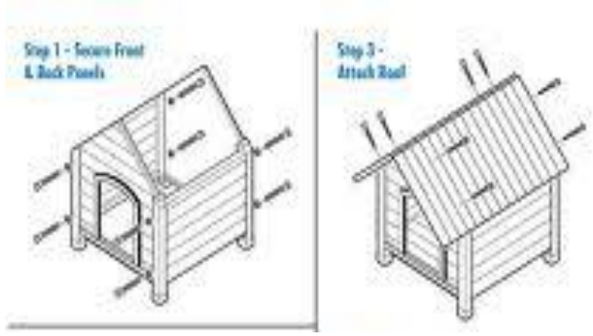
Software Construction - Metaphors

Building Metaphor

Software Development = "Building Software"

- Building Software:
  - Stages of planning, preparation, and execution
  - Vary in kind and degree, e.g. Doghouse vs. Eiffel Tower
  - Reuse vs. Customization
  - Types vs. Methodology

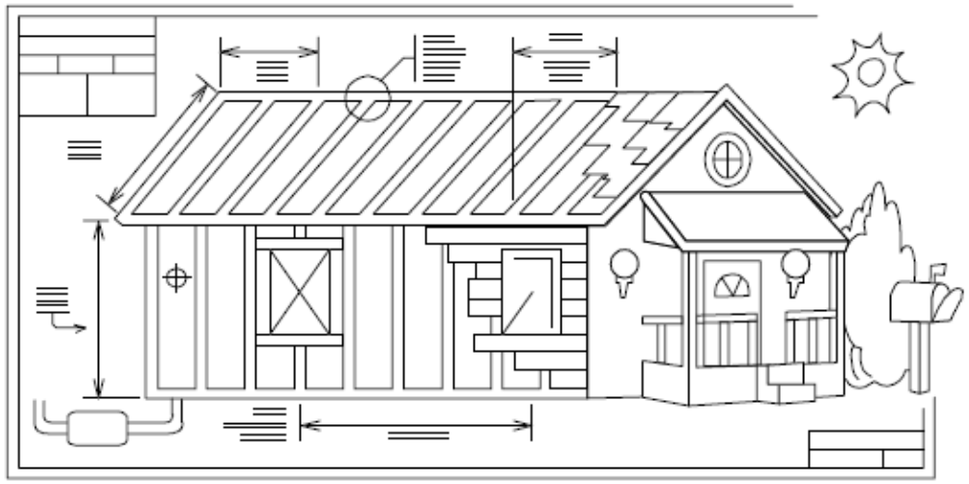




F02xx03

Figure 2-3

The penalty for a mistake on a simple structure is only a little time and maybe some embarrassment.



F02xx04

Figure 2-4

More complicated structures require more careful planning.

Software Construction - Metaphors

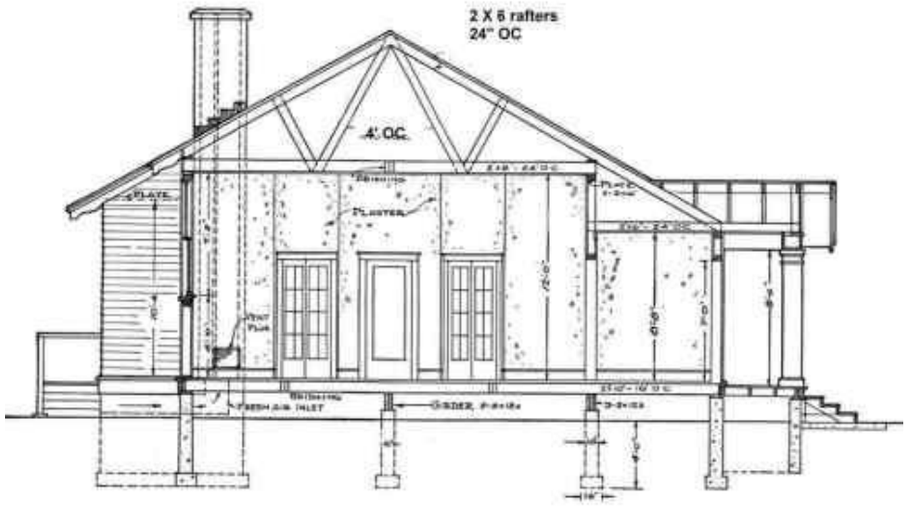
QUESTION

- What kind of things to do during *"House Construction"*?

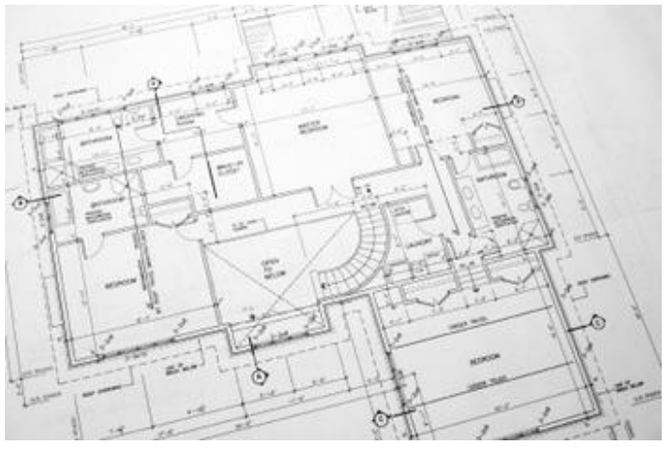
- What kind of things to do during *"Software Construction"*?

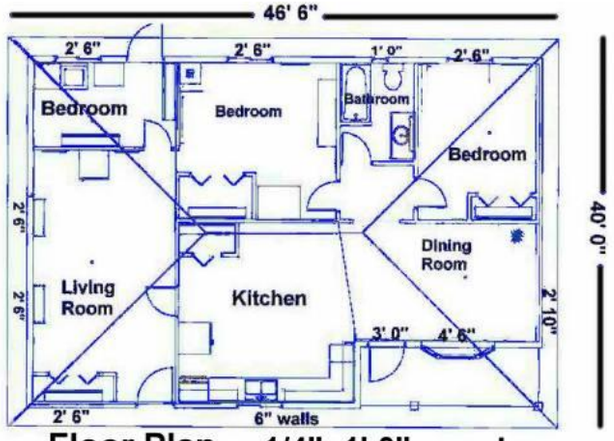
Software Construction - Metaphors

Building A House	Building Software
Decide what kind of a house to build	Problem Definition
Building Architecture Design	Software Architectural Design
Detailed Blueprints	Detailed Software Design
Prepare building site, lay foundation, frame the house, put siding & roof, plumbing & wiring	Software Construction
Painting, decorating, furnishings, landscaping	Software Optimization

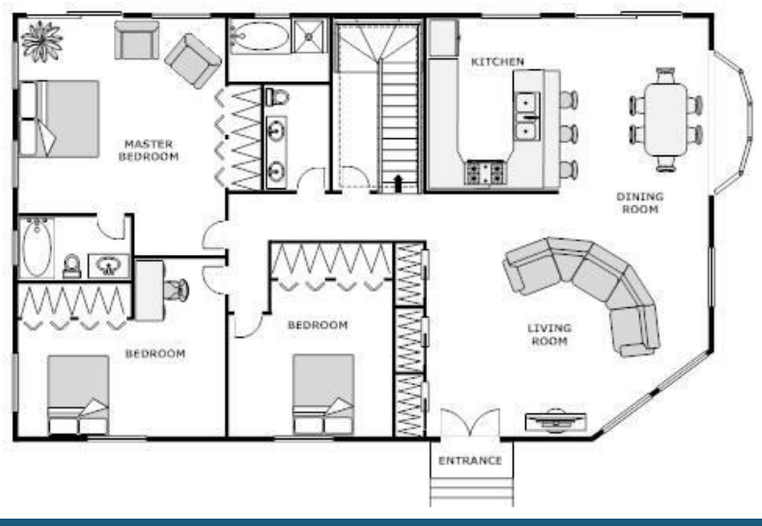


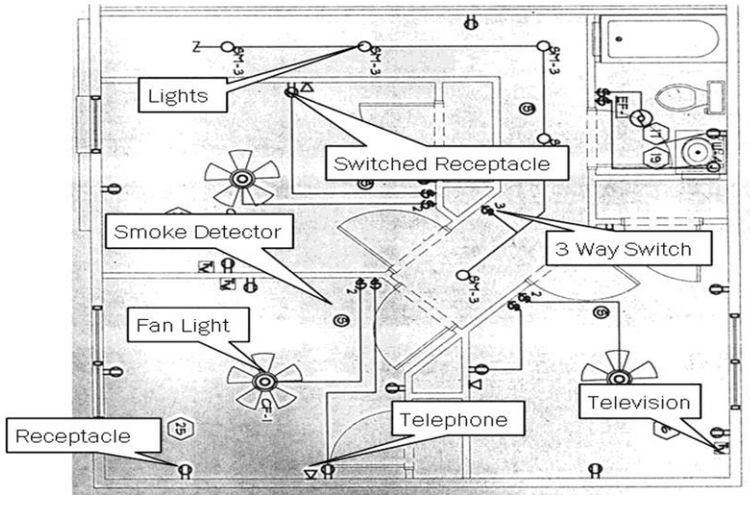
Sections View

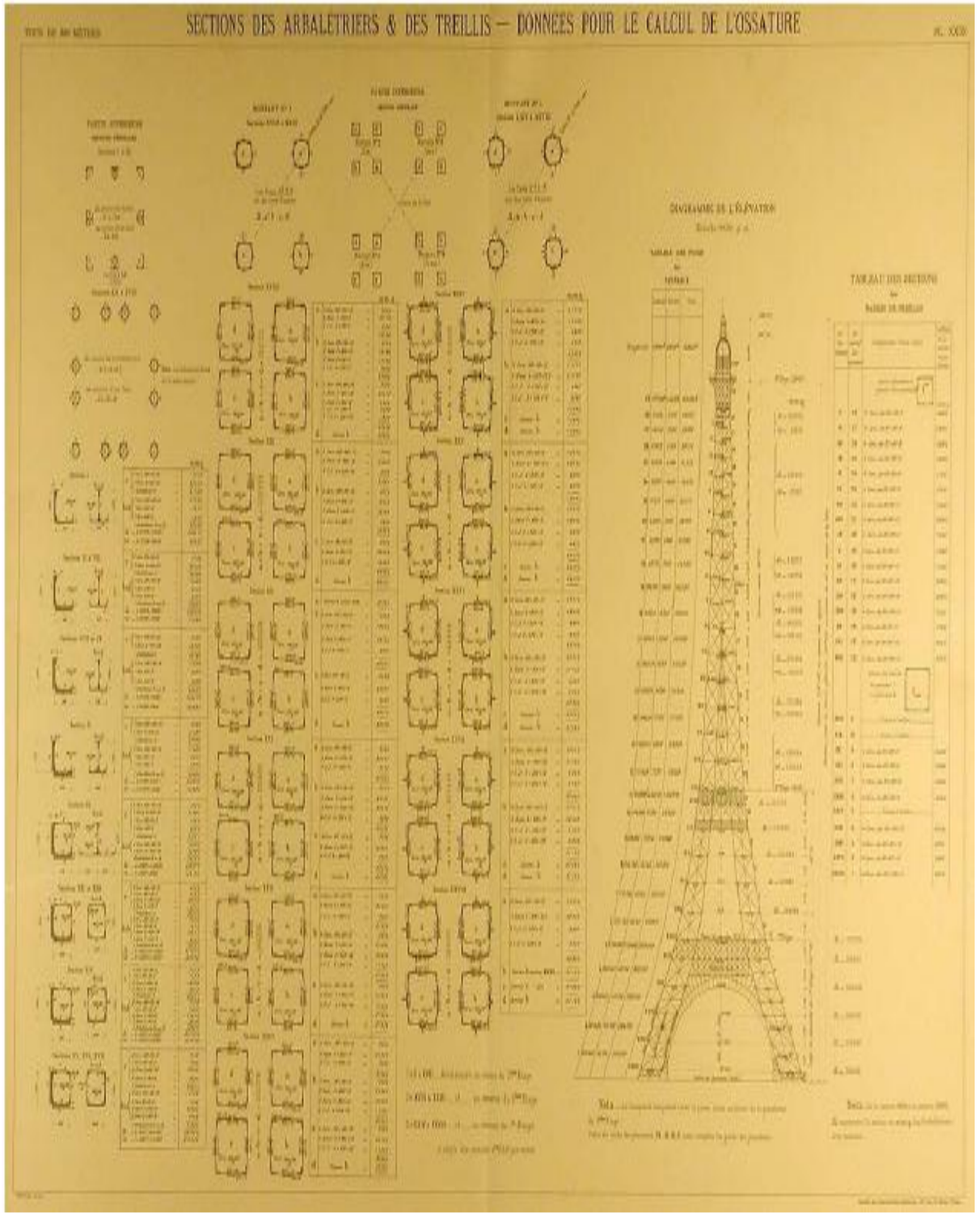




Floor Plan 1/4"=1' 0" scale






SECTIONS DES ARBALÉTRIERS & DES TREILLIS — DONNÉES POUR LE CALCUL DE L'OSSATURE

Software Construction – Metaphors - Complexity

Eiffel Tower

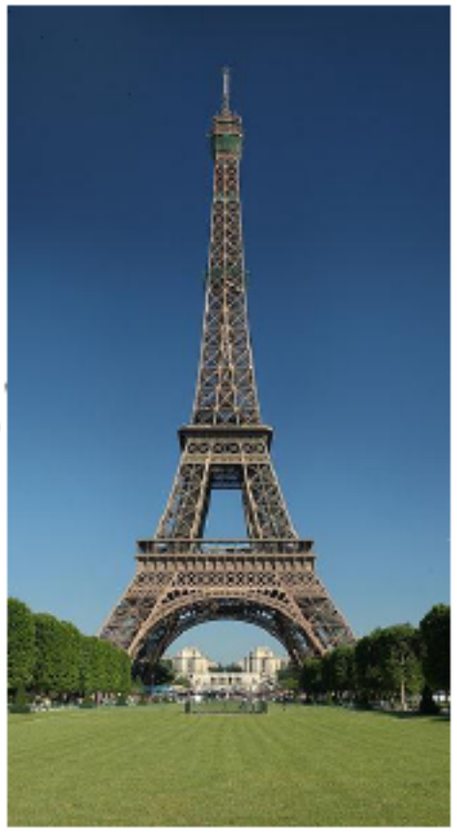
- Studies began: 1884 (324 metres)
- Construction : 1887 – 1889
- Workmen:
  - 50 engineers, designers produced 5,300 "blueprints
  - 326 ironworkers produced 18,038 individual parts
  - 121 men assembling on the construction site

Software Construction – Metaphors - Complexity

Software with 1,000,000 lines of code (LOC)

- 69 kinds of documentation
- Requirements specification: 4,000 – 5,000 pages
- Design documentation: 2-3 times as extensive as the requirements
- Workmen:
  - It takes about 1 month for 1 person to produce 2,000 LOC
  - 500 months (42 years) for 1 person to produce 1,000,000 LOC
  - The project of 42 people to produce this software in 1 year
- Most agile projects can accommodate up to 20 people
- Software for many banking, government and large enterprise applications easily exceed 10,000,000 LOC

Software Construction – Metaphors - Complexity



Eiffel Tower

- Studies began: 1884 (324 metres)
- Construction : 1887 – 1889
- Workmen:
  - 50 engineers, designers produced 5,300 "blueprints"
  - 326 ironworkers produced 18,038 individual parts
  - 121 men assembling on the construction site

Software with 1,000,000 lines of code

- 69 kinds of documentation
- Requirements specification: 4,000-5,000 pages
- Design documentation: 2-3 times as extensive as the requirements

Case Study -- Microsoft's Windows Vista OS

- 50,000,000 lines of code
- 2,000 software developers
- 24 hours to build the system on dual-processor machine
- 50 dependency layers
- 5 years to develop

Software Construction – What?

What is Software Construction?

- ~~By Definition~~
- ~~By Metaphors~~
- By Standard References

Software Construction – Standard References

What is Software Construction?

- ~~By Definition~~
- ~~By Metaphors~~
- By Standard References
  - SE Body of Knowledge (Ref: SWEBOK)
  - International Standard (Ref: ISO29110)
  - Software Life Cycle (Ref: "Code Complete")

Software Construction – Standard References

What is Software Construction?

- ~~By Definition~~
- ~~By Metaphors~~
- By Standard References
  - SE Body of Knowledge (Ref: SWEBOK)
  - ~~International Standard (Ref: ISO29110)~~
  - ~~Software Life Cycle (Ref: "Code Complete")~~

Software Construction - SWEBOK

Software Engineering Body of Knowledge *(Ref:* SWEBOK*)

What is Software Engineering?

- The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software
- 11 Knowledge Areas

---

* ***Software Body of Knowledge***: project founded by IEEE, *www.swebok.org*

Software Construction - SWEBOK

Areas of Knowledge (KA) in Software Engineering (ref: SWEBOK)

1. Software Requirements
2. Software Design
3. Software Construction
4. Software Testing
5. Software Maintenance
6. Software Configuration Management
7. Software Engineering Management
8. Software Engineering Process
9. Software Engineering Tools and Methods
10. Software Quality
11. Knowledge Areas of the Related Disciplines

Software Construction - SWEBOK

11. <u>Areas of Knowledge (KA) in Related Disciplines</u> (ref: SWEBOK)

1. Computer engineering
2. Computer science
3. Management
4. Mathematics
5. Project management
6. Quality management
7. Software ergonomics
8. Systems engineering

Software Construction - SWEBOK

KA 1. **Software Requirements** (ref: SWEBOK)

- Software Requirements Fundamentals
- Software Requirements Process
- Requirements Elicitation
- Requirements Analysis
- Requirements Specification
- Requirements Validation
- Practical Considerations

Software Construction - SWEBOK

KA 2. **Software Design** (ref: SWEBOK)

- Software Design Fundamentals
- Key Issues in Software Design
- Software Structure and Architecture
- Software Design Quality Analysis and Evaluation
- Software Design Notation
- Software Design Strategies and Methods

Software Construction - SWEBOK

KA 3. **Software Construction** (ref: SWEBOK)

- Software Construction Fundamentals
- Managing Construction
- Practical Consideration

Software Construction - SWEBOK

Software Construction – SWEBOK Definition

"The term "software construction" refers to the detailed creation of working, meaningful software through a combination of coding, verification, unit testing, integration testing, and debugging."

Software Construction - SWEBOK

Software Construction – SWEBOK Definition

"The Software Construction Knowledge Area (KA) is linked to all the other KAs, most strongly to Software Design and Software Testing. This is because the software construction process itself involves significant software design and test activity.

It also uses the output of design and provides one of the inputs to testing… Detailed boundaries between design, construction, and testing (if any) will vary depending upon the software life cycle processes that are used in a project."

.

Software Construction - SWEBOK

Software Construction – SWEBOK Definition

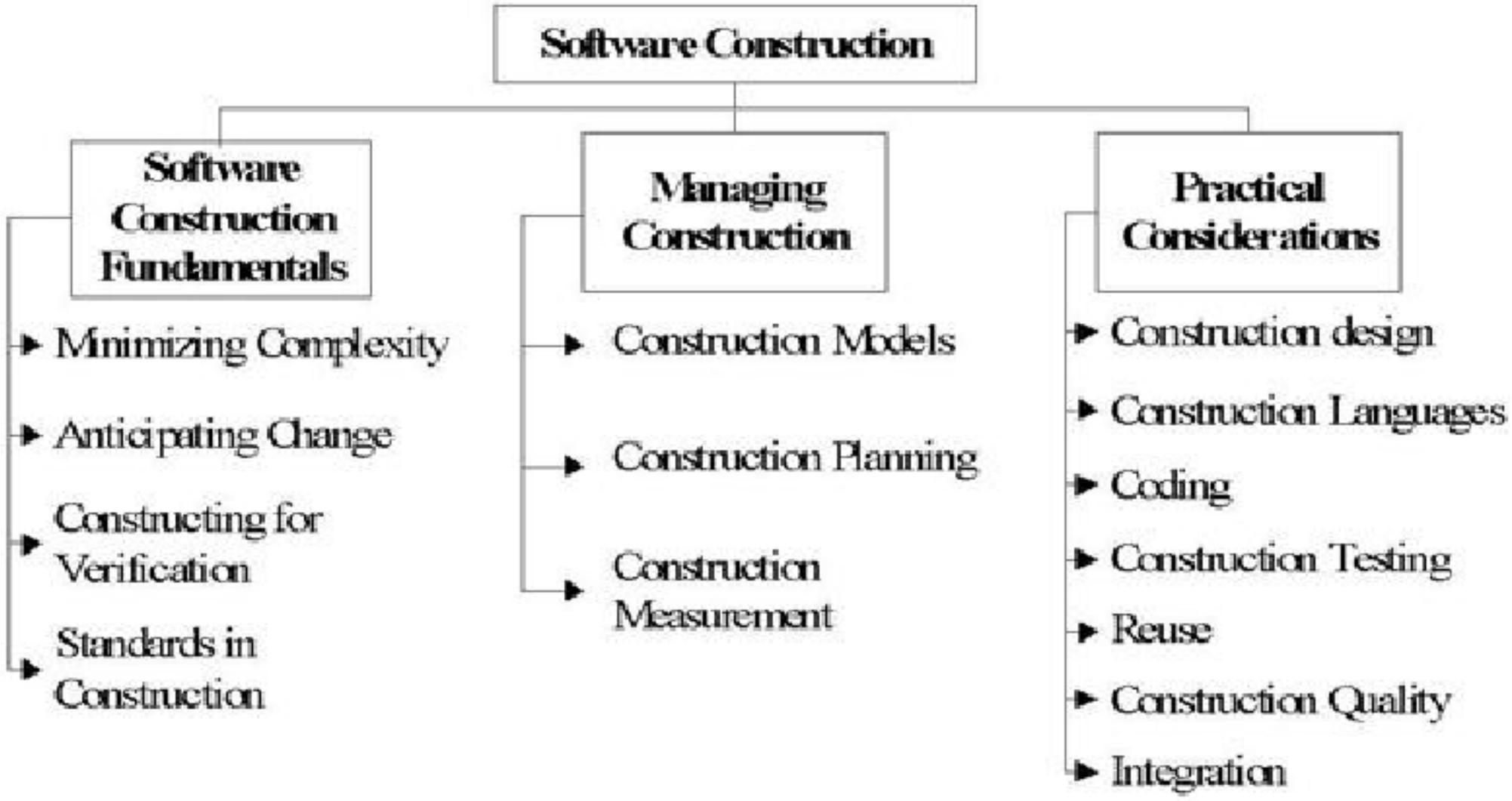


Figure 1. Breakdown of topics for the Software Construction KA.

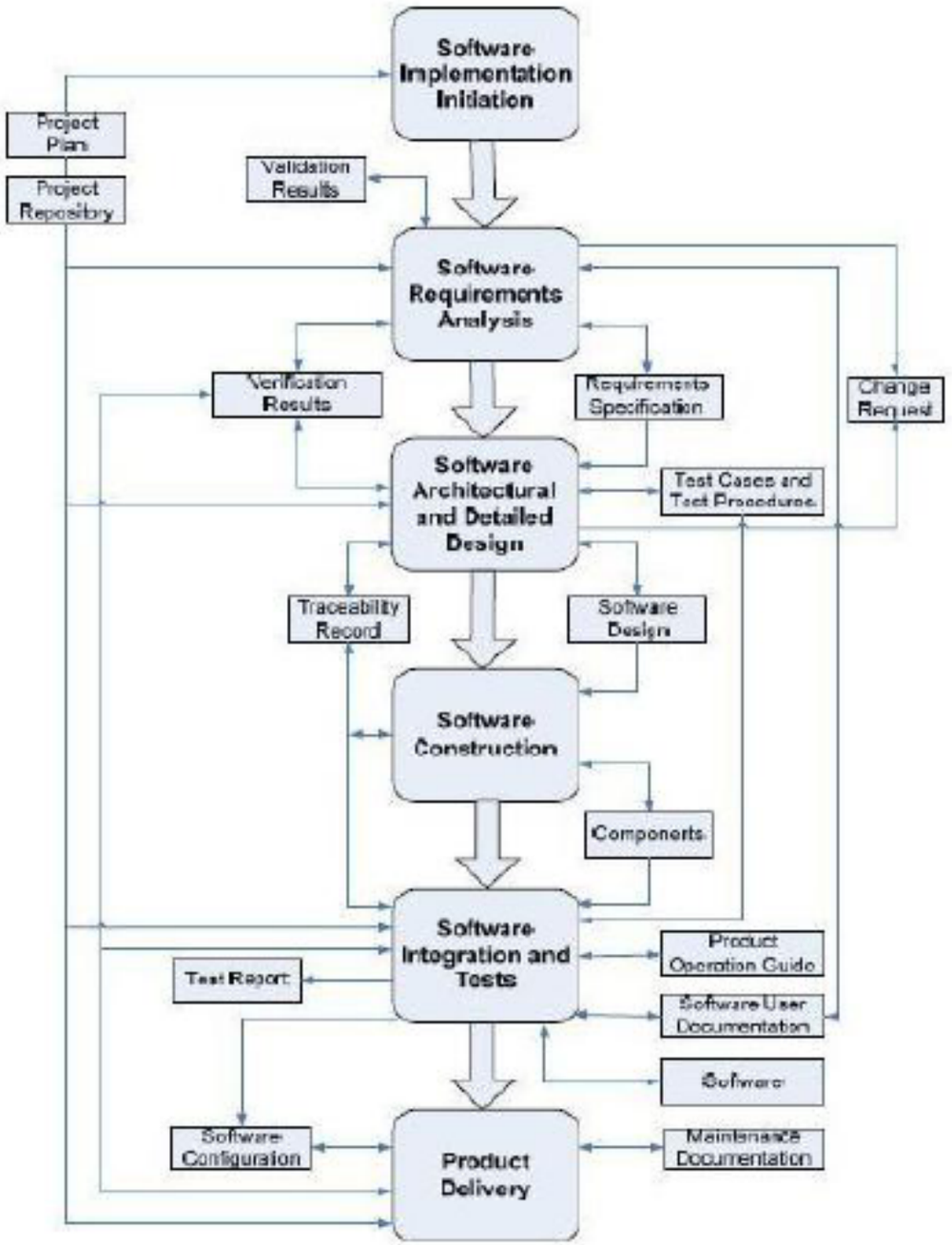
Software Construction – Standard References

What is Software Construction?

- ~~By Definition~~
- ~~By Metaphors~~
- By Standard References
  - ~~Body of Knowledge (Ref: SWEBOK)~~
  - International Standard (Ref: ISO29119)
  - ~~Software Life Cycle (Ref: "Code Complete")~~

Software Construction – Standard

Software Construction – ISO29110 Standard



Software Implementation (SI) Process

SI.1 Software Implementation Initiation

SI.2 Software Requirements Analysis

SI.3 Software Architectural & Detailed Design

SI.4 Software Construction

SI.5 Software Integration & Tests

SI.6 Product Delivery

Software Construction – Standard

Software Construction – ISO29110 Standard

Role	Task List	Input Products	Output Products
TL PR	**SI.4.1 Assign tasks to the Work Team members related to their role, according to the current *Project Plan*.**	***Project Plan***	
PR	**SI.4.2 Understand *Software Design*.**	***Software Design[verified, baselined]***	
PR	**SI.4.3 Construct or update *Software Components* based on the detailed part of the *Software Design*.**	***Software Design [verified, baselined], Traceability Record [verified, baselined]***	***Software Components***
PR	**SI.4.4 Design or update unit test cases and apply them to verify that the *Software Components* implements the detailed part of the *Software Design*.**	***Software Components***	***Software Components[unit tested]***

Software Construction – Standard

Software Construction – ISO29110 Standard

PR	**SI.4.5 Correct the defects found until successful unit test (reaching exit criteria) is achieved.**	***Software Components[unit tested]***	***Software Components[corrected]***
PR	**SI.4.6 Update the *Traceability Record* incorporating *Software Components* constructed or modified.**	***Software Components[corrected]*** ***Traceability Record[verified, baselined].***	***Traceability Record[updated]***
TL	**SI.4.7 Incorporate *Software Components* and *Traceability Record* to the *Software Configuration* as part of the baseline.**	***Software Components[corrected]*** ***Traceability Record[updated]***	***Software Configuration*** ***Software Components[corrected, baselined]*** ***Traceability Record[updated baselined]***

Software Construction – Standard References

What is Software Construction?

- ~~By Definition~~
- ~~By Metaphors~~
- By Standard References
  - ~~Body of Knowledge (Ref: SWEBOK)~~
  - ~~International Standard (Ref: ISO29119)~~
  - Software Life Cycle (Ref: “Code Complete”)

Software Construction - SDLC

Following are activities within the entire process of Software Development (SLDC)*:

1. Problem definition
2. Requirements development
3. Construction planning
4. Software architecture (high level design)
5. Detailed design
6. Coding and debugging
7. Unit testing
8. Integration testing
9. Integration
10. System testing
11. Corrective maintenance

---

* Ref: *Code Complete*, Chapter 1

Software Construction - SDLC

The following activities denoted by * are considered parts of *"Software Construction"*?

1. Problem definition
2. Requirements development
3. Software architecture (high level design)
4. Construction planning*
5. Detailed design*
6. **Coding and debugging***
7. Unit testing*
8. Integration testing*
9. Integration*
10. System testing
11. Corrective maintenance

Software Construction

---

* Ref: *Code Complete*, Chapter 1

Software Construction – Standard Reference

QUESTION

The following 3 references all have defined what the term "*Software Construction*" mean?

- SWEBOK
- International Standard ISO 29110
- Textbook: Code Complete

Though in common in most parts, their definitions of "*Software Construction*" however differ slightly in some details. Identify the areas of agreements and differences among these 3 definitions.

Software Construction
OVERVIEW

OUTLINE

1. ~~*What is software construction?*~~
2. *Why is software construction important?*
3. ~~*How to do software construction?*~~

Software Construction - Why?

2. *Why is software construction important?*

- A large part of software development
  - 30-80 % total time spent on the project
- The central activity in software development
- Programmer's productivity can improve
  - 10-20 times difference between individual programmers
- Construction's product is often the only an accurate description of the software
  - Source code may be the only up-to-date information available
- Guaranteed to be done
  - Even any imperfect real world projects

Software Construction
OVERVIEW

OUTLINE

1. ~~*What is software construction?*~~
2. ~~*Why is software construction important?*~~
3. *How to do software construction?*

Software Construction – How?

Software Construction

3. *How to do software construction?*

Major groups of tasks involved in software construction* :

1. Planning for the construction
2. Doing the construction
3. Documenting the construction
4. Measuring & fine-tuning the construction

---

* ***All the above topics will be covered later in this course.***

Software Construction – How?

Detailed tasks to be carried out during *"Software Construction"*?

- Verifying that the groundwork has been laid so that construction can proceed
- Determining how your code will be tested
- Designing and writing classes and routines
- Creating and naming variables and named constants
- Selecting control structures and organizing blocks of statements
- Unit testing, integration testing, and debugging your own code
- Reviewing other team members' low-level designs and code and having them review yours
- Polishing code by carefully formatting and commenting it
- Integrating software components that were created separately
- Tuning code to make it faster and use fewer resources

_____

* Ref: *Code Complete*, Chapter 1

Software Construction – Standard

Software Construction – ISO29110 Standard

Role	Task List	Input Products	Output Products
TL PR	**SI.4.1 Assign tasks to the Work Team members related to their role, according to the current *Project Plan*.**	***Project Plan***	
PR	**SI.4.2 Understand *Software Design*.**	***Software Design[verified, baselined]***	
PR	**SI.4.3 Construct or update *Software Components* based on the detailed part of the *Software Design*.**	***Software Design [verified, baselined], Traceability Record [verified, baselined]***	***Software Components***
PR	**SI.4.4 Design or update unit test cases and apply them to verify that the *Software Components* implements the detailed part of the *Software Design*.**	***Software Components***	***Software Components[unit tested]***

Software Construction – Standard

Software Construction – ISO29110 Standard

PR	**SI.4.5 Correct the defects found until successful unit test (reaching exit criteria) is achieved.**	***Software Components[unit tested]***	***Software Components[corrected]***
PR	**SI.4.6 Update the *Traceability Record* incorporating *Software Components* constructed or modified.**	***Software Components[corrected]*** ***Traceability Record[verified, baselined].***	***Traceability Record[updated]***
TL	**SI.4.7 Incorporate *Software Components* and *Traceability Record* to the *Software Configuration* as part of the baseline.**	***Software Components[corrected]*** ***Traceability Record[updated]***	***Software Configuration*** ***Software Components[corrected, baselined]*** ***Traceability Record[updated baselined]***

Software Construction – How?

Works needed in software development but NOT parts of *"Software Construction"*

- Management
- Requirements Development
- Software Architecture
- User-interface Design
- System Testing
- Maintenance

---

* Each of these activities are as important as those construction activities ---at least to the success of any project of more than 1 or 2 people and last longer than a few weeks.

CHAPTER SUMMARY
Chapter 01 – Overview of Software Construction

KEY POINT

- *What is Software Construction?*

MAJOR TOPICS

1. Definition – General meaning of *"construction"*
2. Metaphors – *The construction of software is like….*
    - Writing
    - Growing
    - Building
3. Standard References – Technical definition of *"software construction"*
    - SWEBOK
    - ISO 29110
    - Code Complete

QUESTION

- Compare and contrast the definitions of *"software construction"* according to the 3 references.

Software Construction - Homework

REFERENCE

Textbook: *"Code Complete"*, second edition

- Chapter 1: *Welcome to Software Construction*
- Chapter 2: *Metaphors for Richer Understanding of Software Development*