Software Engineering Principles

### 28/04/2025

Joanna Batstone

VP, IBM Watson

The importance of learning:

* New technologies
* Cloud
* Artificial Intelligence
* Financial Services / Healthcare
* Apply technology

The importance of software engineering principles / architecture:

* Translate ideas that you may have (workflows, design, scripts) into code

The importance of software development process:

* Design Thinking
  + Who are users? Stakeholders? Experience of people who’s using.
* User Experience

### **📚 Key Learning Outcomes:**

By the end of the subject, you should be able to:

* **Apply core software engineering techniques** to real-world problems.
* **Author professional documents** like formal specs, requirement docs, and test plans.
* **Design, develop, and evaluate** software systems across the SDLC.
* **Deliver high-quality software** solutions that meet user and stakeholder needs.
* **Demonstrate ethical and legal responsibility** in software development.

### **💡 Topics You’ll Explore:**

* Requirements engineering
* Software modelling and design
* Software architecture
* Verification and validation
* Quality assurance
* Ethical standards in engineering practice

### **Introduction:**

The real complexities in software development are determining what the software should do; how to structure it to simplify implementation; managing the development process; testing to ensure quality; maintaining the software; and documenting the software so that users can understand.

### 

### **Books:**

Stephens, R. (2015). *Beginning Software Engineering*. Retrieved from [https://ebookcentral-proquest-com.torrens.idm.oclc.org/lib/think/reader.action?ppg=35&docID=1895174&tm=1542676933113 pp 3-13](https://ebookcentral-proquest-com.torrens.idm.oclc.org/lib/think/reader.action?ppg=35&docID=1895174&tm=1542676933113%20pp%203-13)

All software engineering projects must handle the same basic tasks, although different development models may approach these tasks in various ways.

Here are important **concepts** discussed in this chapter**: SUMMARY** and why they are important

1. **Requirements Gathering**: This is a fundamental task in software engineering where the needs and expectations of stakeholders are identified and documented. Its significance lies in ensuring that the final product meets the users' needs and reduces the risk of project failure due to misunderstandings.
2. **Development**: This refers to the process of writing and maintaining the source code of software applications. It is significant as it transforms the design and requirements into a functional product, directly impacting the quality and performance of the software.
3. **Testing**: Testing is the process of evaluating a system or its components to determine whether they meet the specified requirements. Its importance is highlighted in the document as it helps catch bugs early, ensuring a more reliable and robust final product.
4. **Deployment**: Deployment is the stage in software engineering where the developed software is made available for use. This task is crucial as it involves the transition of the software from a development environment to a live environment, impacting user experience and satisfaction.
5. **Maintenance**: Maintenance involves the ongoing support and updates of software after its deployment to fix issues and improve performance. Its significance is underscored in the document as it ensures the longevity and relevance of the software in a changing environment.