Introduction to Software Engineering

Lecture 1

Course Objectives

- Understanding the software engineering process and different process models and how to choose between them.
- How to elicit requirements from a client and specify them.
- Design in the large, including principled choice of a software architecture.
- Develop a system design using UML notation.
- Understanding good coding practices, including documentation, contracts, regression tests and daily builds.
- Various quality assurance techniques, including unit testing, functional testing, and automated analysis tools.
- Understanding of the role of project management including planning, scheduling, risk management, etc.

Contribution to Program Student Outcomes

- (SO1) - Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.
- (SO2) - Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline.
- (SO3) - Communicate effectively in a variety of professional contexts.
- (SO4) - Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.
- (SO5) - Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline

Course Outline

- **Week 1:** Introduction to Software Engineering Course
- **Week 2:** Software Process
- **Week 3:** Software Process Models
- **Week 4:** Software Requirements Gathering
- **Week 5:** Software Requirements Analysis and Specification
- **Week 6:** Structured Analysis
- **Week 7:** 7th Week Examination
- **Week 8:** Object Oriented Analysis and Design
- **Week 9:** Design Concepts and Principles
- **Week 10:** Architectural Design
- **Week 11:** Implementation Coding Practices and Principles
- **Week 12:** Software Quality Assurance
- **Week 13:** Verification and Validation
- **Week 14:** Project Management
- **Week 15:** Revision
- **Week 16:** Final Examination

Core Topics of Software Engineering

- According to Software Engineering 2014, a set of curriculum guidelines developed by the Joint Task Force on Computing Curricula, the IEEE Computer Society, and the Association for Computing Machinery (ACM), studies in software engineering will focus on certain core topics, including:
  - Computing essentials;
  - Mathematical and engineering fundamentals;
  - Professional practice;
  - Software modeling and analysis;
  - Requirements analysis and specification;
  - Software design;
  - Software verification and validation;
  - Software process;
  - Software quality; and security.

Course Book Information

- **Text Book**:

  Ian Summerville, ninth Edition, *Software Engineering*, Addison Wesley.

- **Web**:

  http://www.cs.st-andrews.ac.uk/~ifs/Books/SE8/Syllabuses/index.html

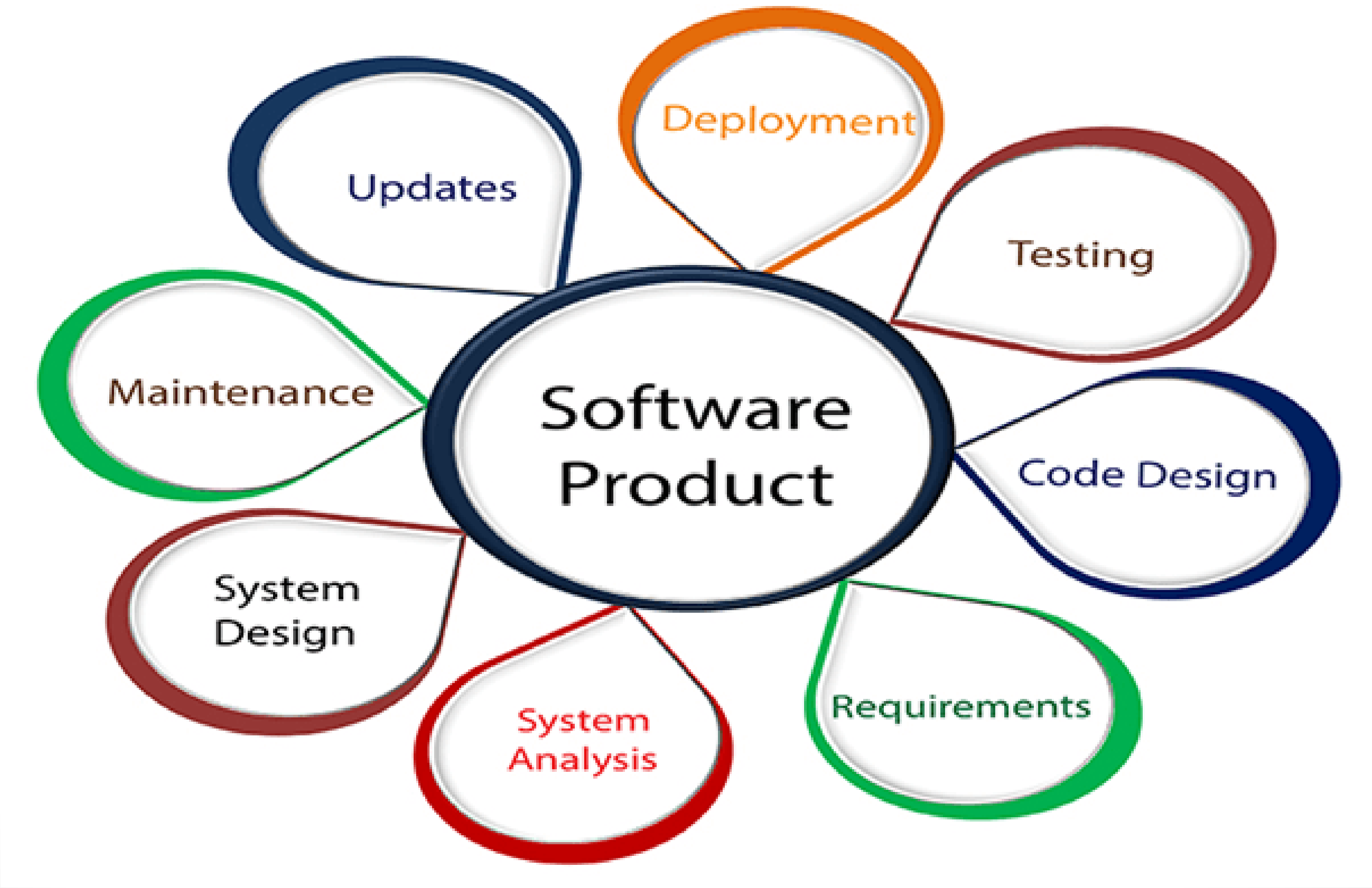
Grading Policy

Assessment	Method
7th Week (30%)	Exam (20) + Section Quiz (5) + Presentation (5)
12th Week (20%)	Project (20)
Coursework (10%)	Assignments (5) + Section Quiz (5)
Final (40%)	Examination (40)

General Policies

- Attendance will be taken every class and will be reported.
- Official excuse for any class absence must be presented **within one week** after that class.
- Students are expected to regularly check the course website on www.aast.edu for lecture notes and updates.
- It is important that any work submitted is your own. Plagiarism and/or collusion will result in a possible disciplinary action.
- All group members should attend the final project discussion with the course teaching assistant Ms. Hager Hussein. Absence of any group member will be taken with penalty.
- Any late submissions will be taken with penalty.
- No coursework will be accepted after the 15th week, not even with penalty.

What is Software Engineering?



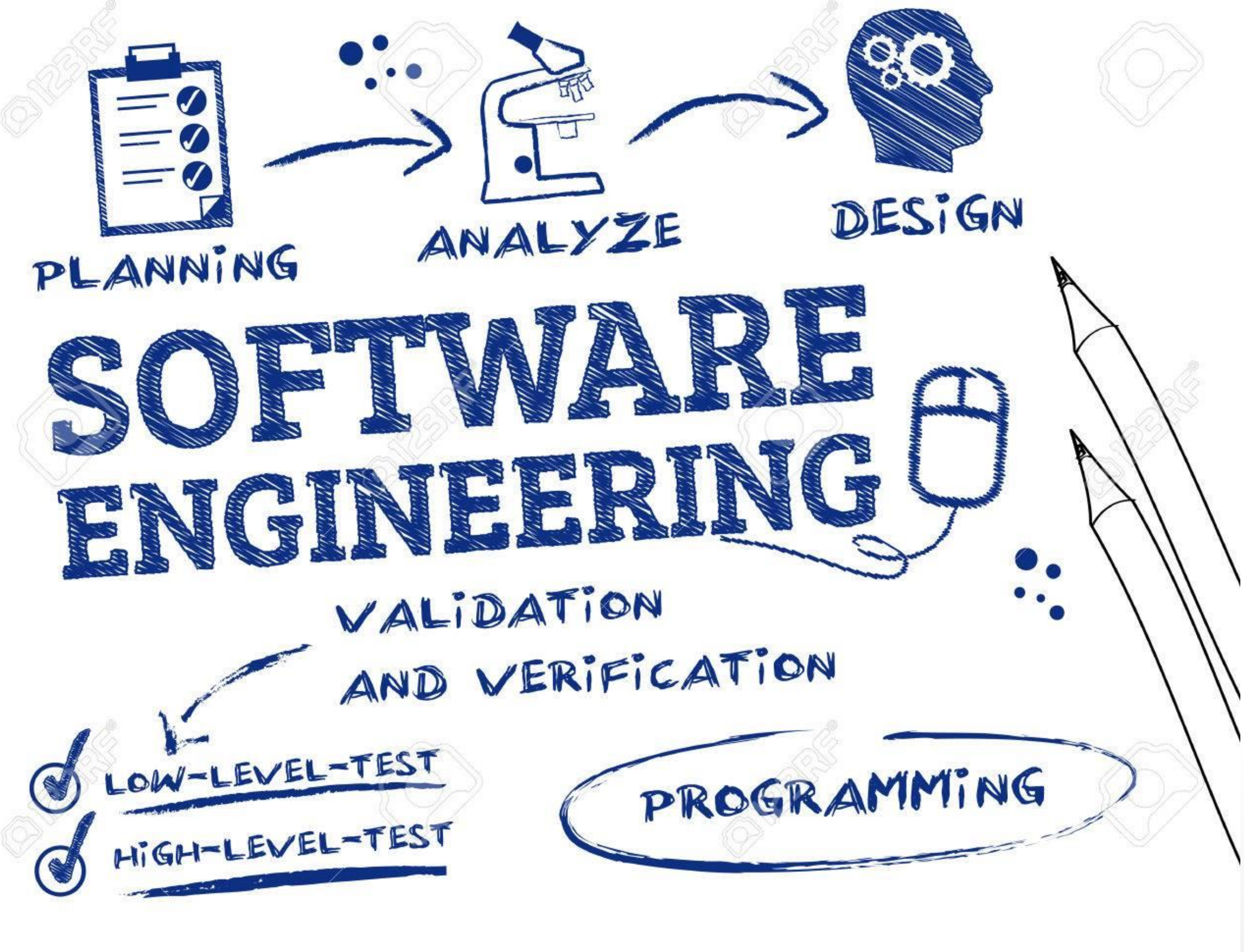
What is Software Engineering?

- The term **software engineering** is the product of two words, **software**, and **engineering**.
- The **software** is a collection of integrated programs.
- Software subsists of carefully-organized instructions and code written by developers on any of various particular computer languages.
- Computer programs and related documentation such as requirements, design models and user manuals.
- **Engineering** is the application of **scientific** and **practical** knowledge to **invent, design, build, maintain**, and **improve frameworks, processes, etc**.

What is Software Engineering?

- **Software Engineering** is an engineering branch related to the evolution of software product using systematic well-defined scientific principles, techniques, and procedures.
- The result of software engineering is an effective and reliable software product.

What is Software Engineering?



Why is Software Engineering required?

Software Engineering is required due to the following reasons:

- To manage Large software
- For more Scalability
- Cost Management
- To manage the dynamic nature of software
- For better quality Management

Why is Software Engineering required?

The necessity of software engineering appears because of a higher rate of progress in user requirements and the environment on which the program is working.

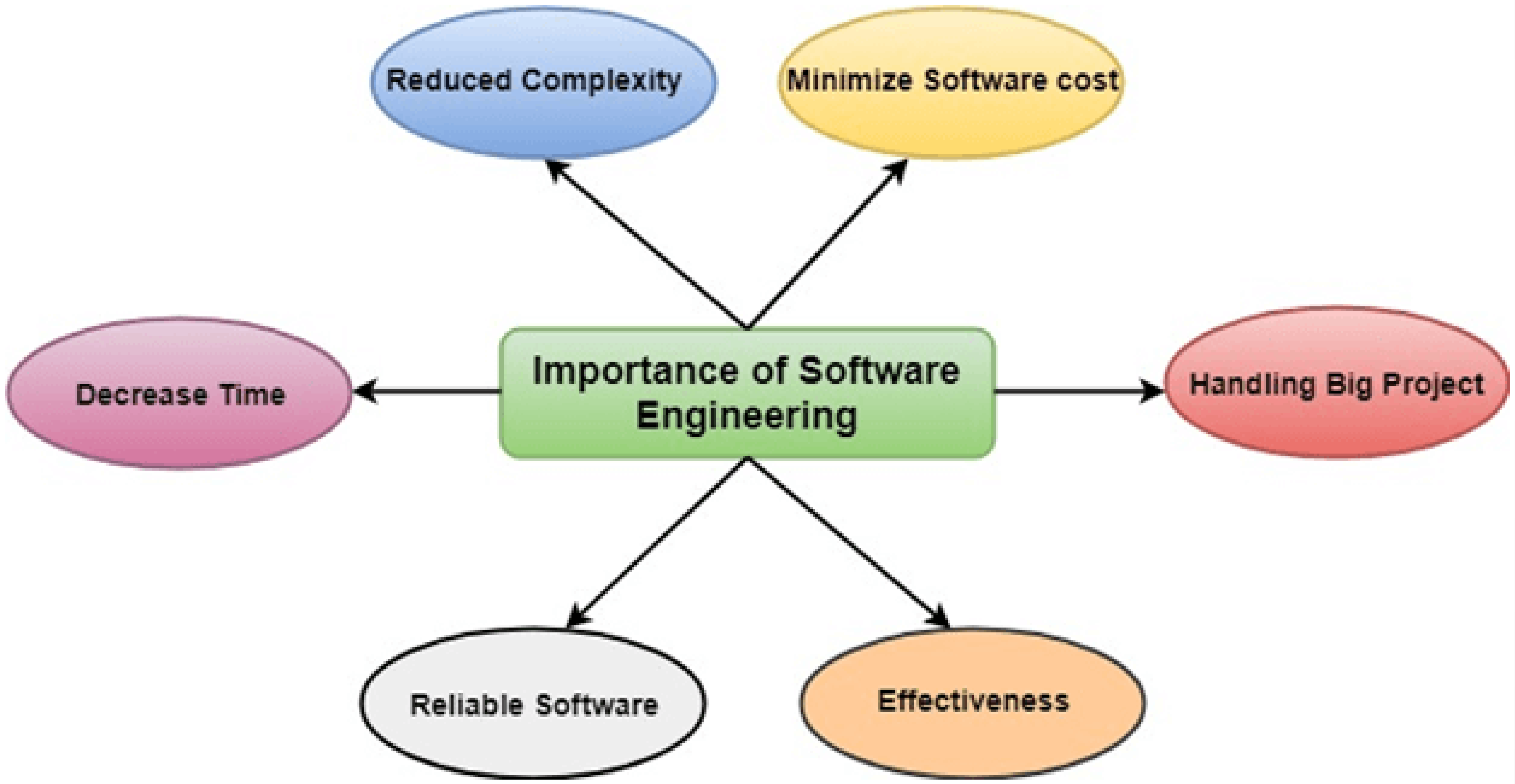
- **Huge Programming:** It is simpler to manufacture a wall than to a house or building, similarly, as the measure of programming become extensive engineering has to step to give it a scientific process.
- **Adaptability:** If the software procedure were not based on scientific and engineering ideas, it would be simpler to re-create new software than to scale an existing one.
- **Cost:** As the hardware industry has demonstrated its skills and huge manufacturing has let down the cost of computer and electronic hardware. But the cost of programming remains high if the proper process is not adapted.
- **Dynamic Nature:** The continually growing and adapting nature of programming hugely depends upon the environment in which the client works. If the quality of the software is continually changing, new upgrades need to be done in the existing one.
- **Quality Management:** Better procedure of software development provides a better and quality software product.

Characteristics of a good software engineer

The features that good software engineers should possess are as follows:

- Exposure to systematic methods, i.e., familiarity with software engineering principles.
- Good technical knowledge of the project range (Domain knowledge).
- Good programming abilities.
- Good communication skills. These skills comprise of oral, written, and interpersonal skills.
- High motivation.
- Sound knowledge of fundamentals of computer science.
- Ability to work in a team
- Discipline

Importance of Software Engineering



Importance of Software Engineering

(1) Reduces complexity:

- Large software systems are always complicated and challenging to progress.
- Software engineering divides big problems into various small issues. And then start solving each small issue one by one.
- All these small problems are solved independently to each other and then integrated together to produce the software product.

(2) To minimize software cost:

- A lot of resources are required to develop large-scale software systems, such as: manpower, software licenses, hardware…etc.
- As companies seek to build cutting-edge software to drive growth, determining the overall budget becomes very tricky.
- Software engineering provides systematic means for having regular interaction and obtaining a budget estimates.

Importance of Software Engineering

(3) To decrease time:

- One of the main criteria to measuring project success and yet the most challenging is delivering software projects on time.
- Schedule issues are the main reason for conflicts on projects, especially during the second half of projects where the actual implementation takes place to produce the final working product.
- Software engineering involves the processes required to ensure timely completion of a project.

(4) Handling Big projects:

- Big projects are not done in a couple of days, and they need lots of work, planning, and management. And to invest six and seven months of any company, it requires heaps of planning, direction, testing, and maintenance.
- Companies provide many resources to the plan to be completed.
- So to handle a big project without any problem, the company has to go for software engineering methods.

Importance of Software Engineering

(5) Reliable software:

- Software reliability is the “probability that the software will execute for a particular period of time without failure”.
- Software engineering provides models for software quality measurements and evaluations. Examples include models for estimating defects inserted and removed throughout the software lifecycle.

(6) Effectiveness:

- Effectiveness means getting the desired results/ doing the right thing!
- From example: achieving missions and goals; generating satisfied customers; producing work of high quality.
- This includes proper use of: communication, technology, organizational and individual knowledge, and resources.

Software Engineering vs. Computer Science

- Computer science is concerned with theory and fundamentals; This field involves the understanding and application of both abstract and concrete knowledge.
- Software engineering is a field largely concerned with the application of engineering processes to the creation, maintenance, and design of software for a variety of different purposes.

Software Engineering vs. System Engineering?

- System engineering is concerned with all aspects of computer-based systems development including hardware, software and process engineering.
- System engineers are involved in system specification, architectural design, integration and deployment.
- Software engineering is part of this process concerned with developing the software infrastructure, control, applications and databases in the system.
- Systems engineering is older than Software Engineering:
  - Complex industrial systems such as trains and chemical plants.
- As the percentage of software systems has increased, software engineering techniques are finding their way into systems engineering.

Employment

- **Specialist IT firms** - such as IT consultancies, large IT providers, software development, internet providers and training firms; organizations that use IT software, systems and equipment, including retailers, law firms, business intelligence and market research organizations, education providers, the armed forces, the public sector and voluntary sector organizations.
- **Manufacturing industry** - including automotive, navigation, telecommunications, manufacturing and construction companies.
- **Financial services** - including global investment banks, financial/banking organizations, security market specialists and the pensions sector.
- **Public utilities** - covering energy and water supply, energy extraction and transport.

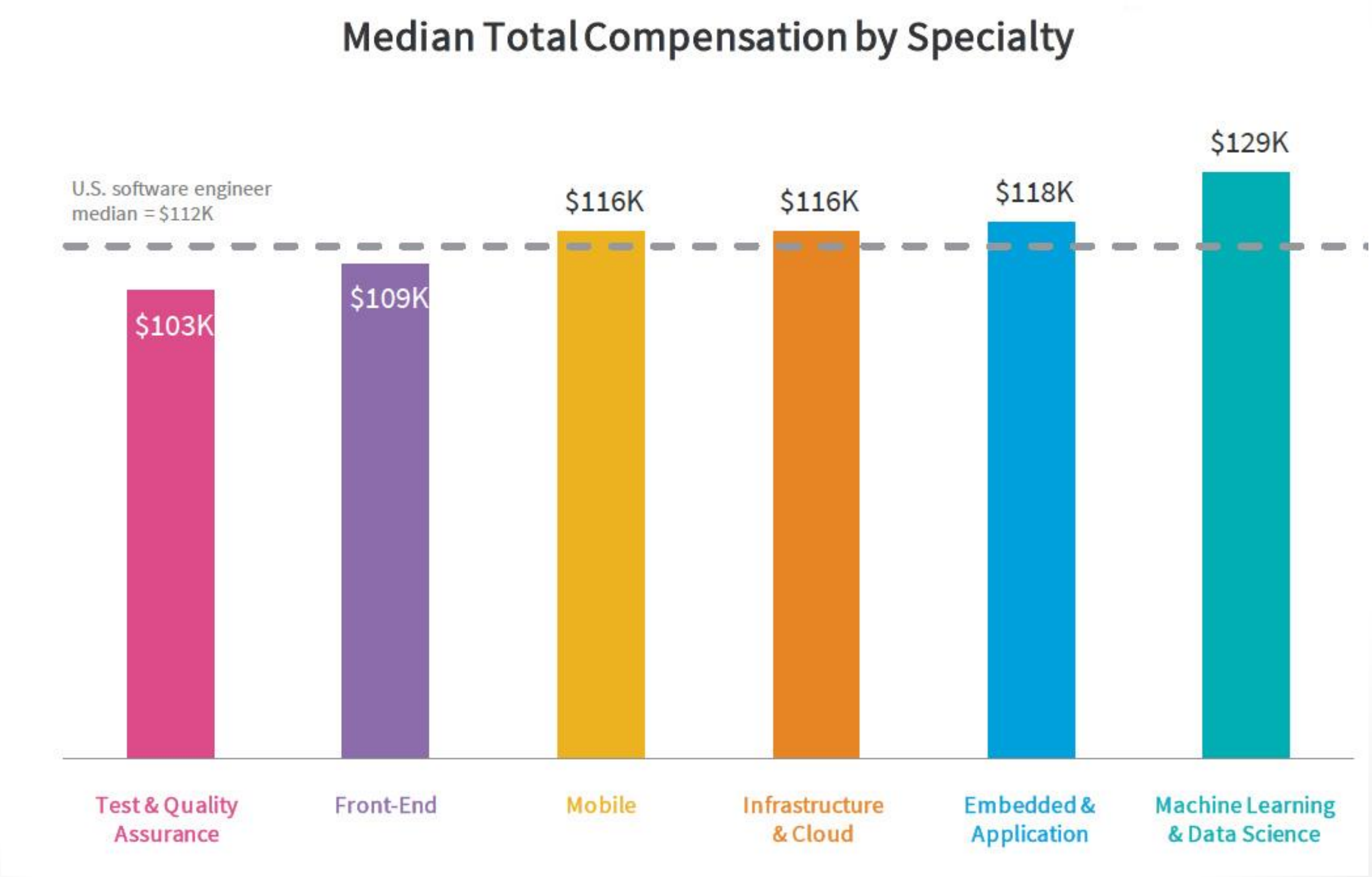
Software Engineer Job Duties

- Analyzing user requirements
- Testing code, refining and rewriting it as necessary
- Researching and designing new software programs
- Developing existing programs by analyzing and identifying areas for modification
- Integrating existing software products and getting incompatible platforms to work together
- Creating technical specifications
- Writing operational documentation with technical authors
- Maintaining systems by monitoring and correcting software defects

Software Engineer Job Duties

- Working closely with other staff, such as: project managers, graphic artists, user experience (UX) designers, other developers, systems analysts and sales and marketing professionals.
- Consulting clients and colleagues concerning the maintenance and performance of software systems with a view to writing or modifying current operating systems
- Investigating new technologies

Recent Facts About Software Engineer Job Opportunities



Recent Facts About Software Engineer Job Opportunities

After examining millions of LinkedIn profiles of software engineers (Feb 2018), its was found that most engineers fit into six areas:

- **Test and Quality Assurance Engineers:** The most gender-diverse group. Women represent only around 30 percent of its talent pool.
- **Mobile Engineers:** The second-biggest specialty, and they're also the youngest cohort – 37 percent of these engineers have less than 10 years of experience, more than any other specialty.
- **Front-End Engineers:** By far the biggest talent pool, their population is more than double that of the second-biggest specialty.
- **Infrastructure and Cloud Computing Engineers:** 77 percent of infrastructure and cloud professionals have more than 10 years of experience.
- **Embedded and Application Engineers:** These engineers focus primarily on the IoT and are one of the smallest talent pools, but also the most experienced. It's also the least gender-diverse.
- **Machine Learning and Data Science Engineers:** The smallest specialty – and the most in-demand.

Questions?