FACULDADE ENGENHARIA
Departamento de Informática

Engenharia de Software (14341, 16230, 15386)

Lab#1 - Introduction

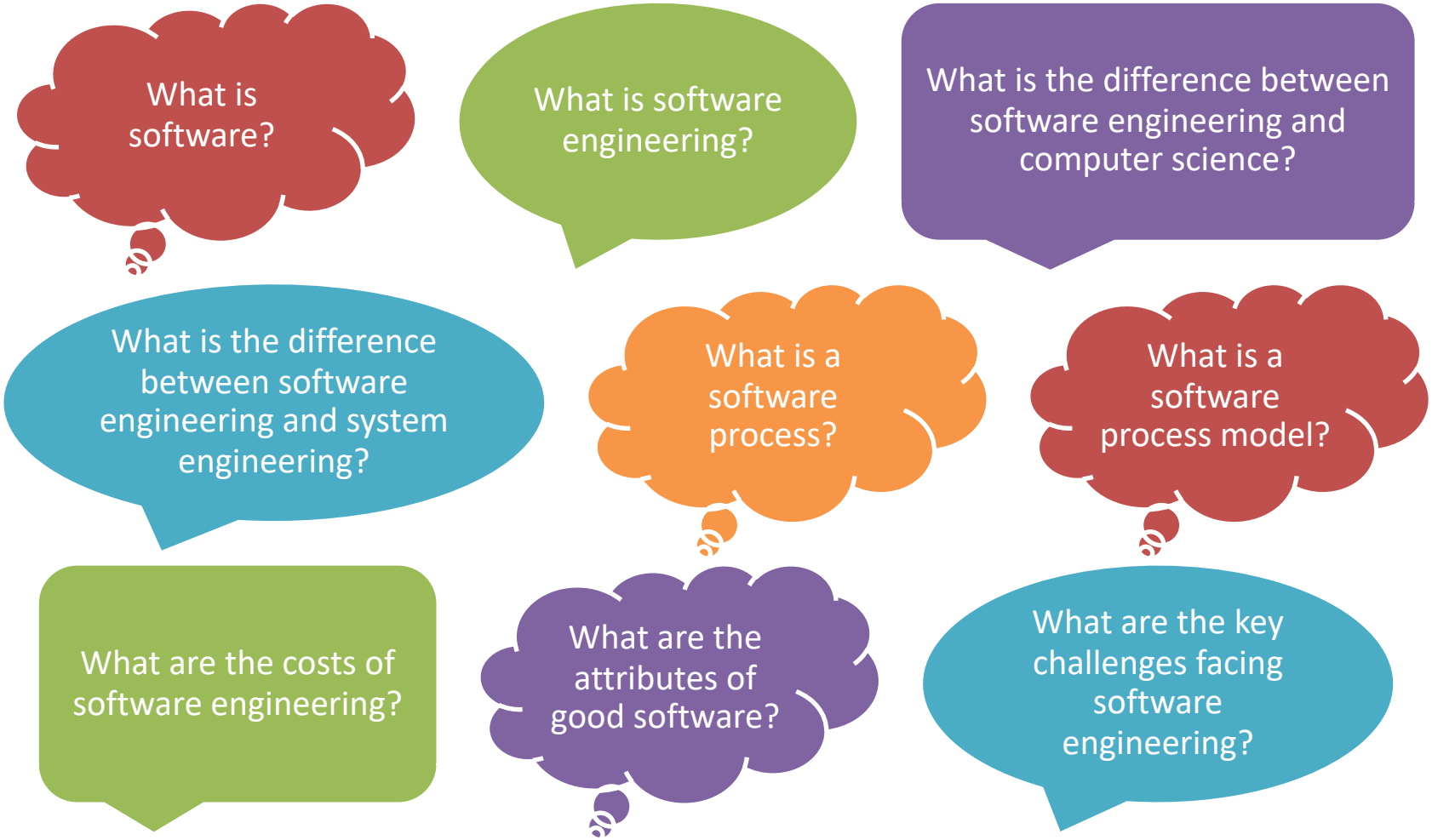
(adapted from lecture notes of Professor Sofia Ouhbi, Uppsala University,
Software Engineering: International Version, Ian Sommerville, Pearson, 2015, and
Engineering Software Products: An Introduction to Modern Software Engineering, Ian Sommerville, Pearson, 2020)

Software engineering

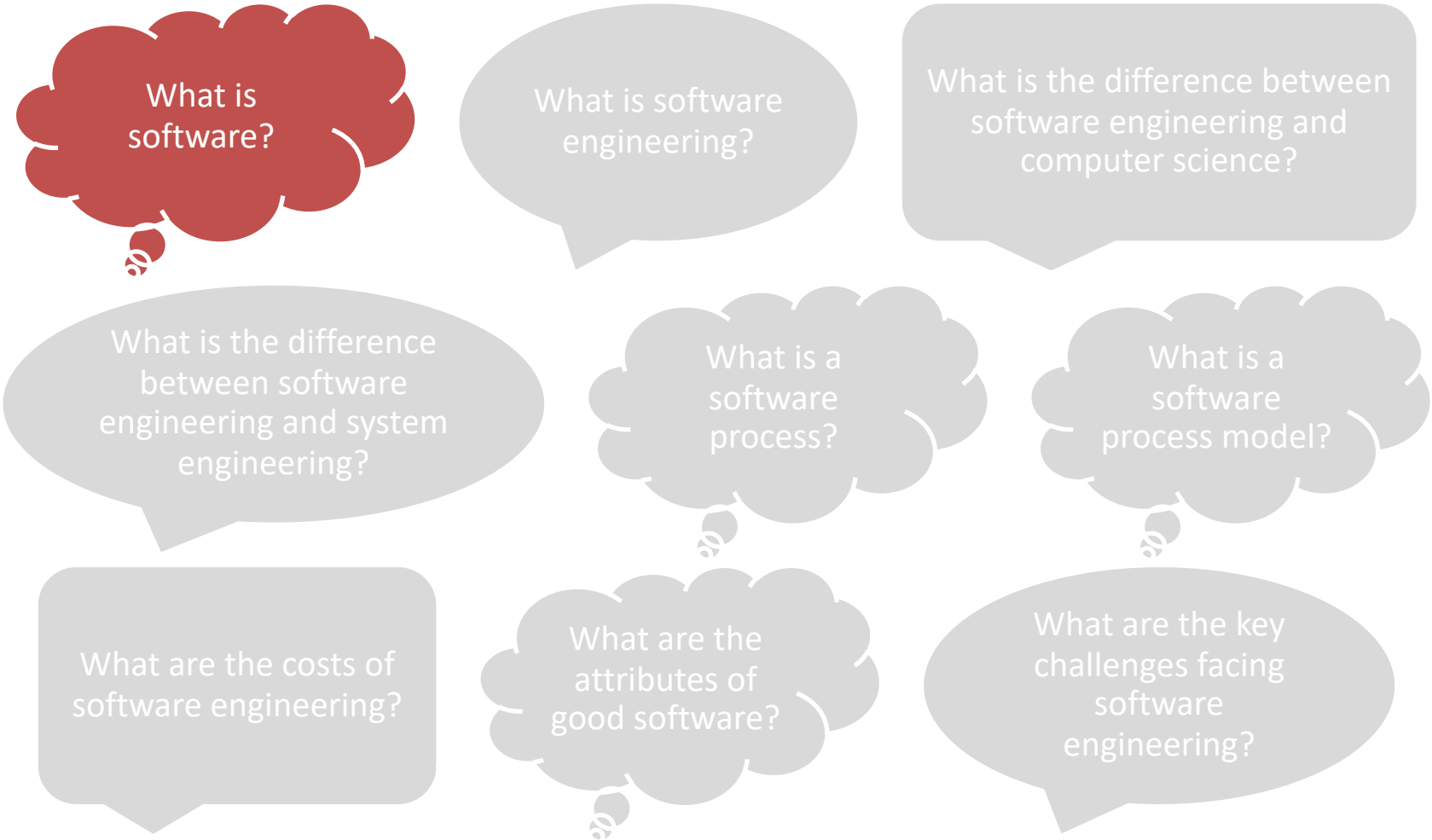
- The economies of ALL developed nations are dependent on software.
- More and more systems are software controlled.
- Software engineering is concerned with theories, methods and tools for professional software development.
- Expenditure on software represents a significant fraction of GNP in all developed countries.

Professional software development

Frequently asked questions about software engineering



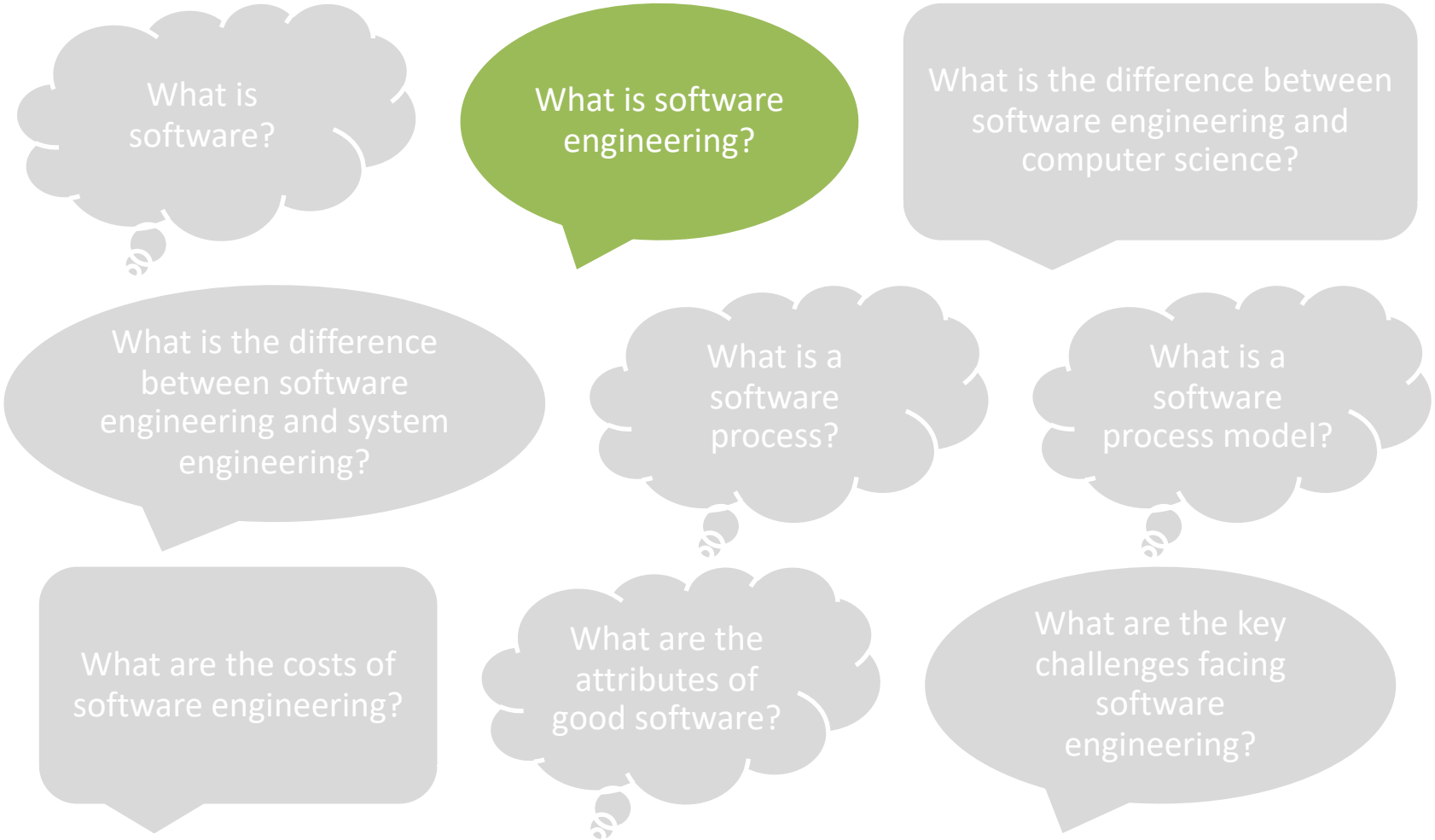
Frequently asked questions about software engineering



Software

- Computer programs and associated documentation. Software products may be developed for a particular customer or may be developed for a general market.
- Generic products
  - Stand-alone systems that are marketed and sold to any customer who wishes to buy them.
  - Examples – PC software such as graphics programs, project management tools; CAD software; software for specific markets such as appointments systems for dentists.
- Customized products
  - Software that is commissioned by a specific customer to meet their own needs.
  - Examples – embedded control systems, air traffic control software, traffic monitoring systems.

Frequently asked questions about software engineering



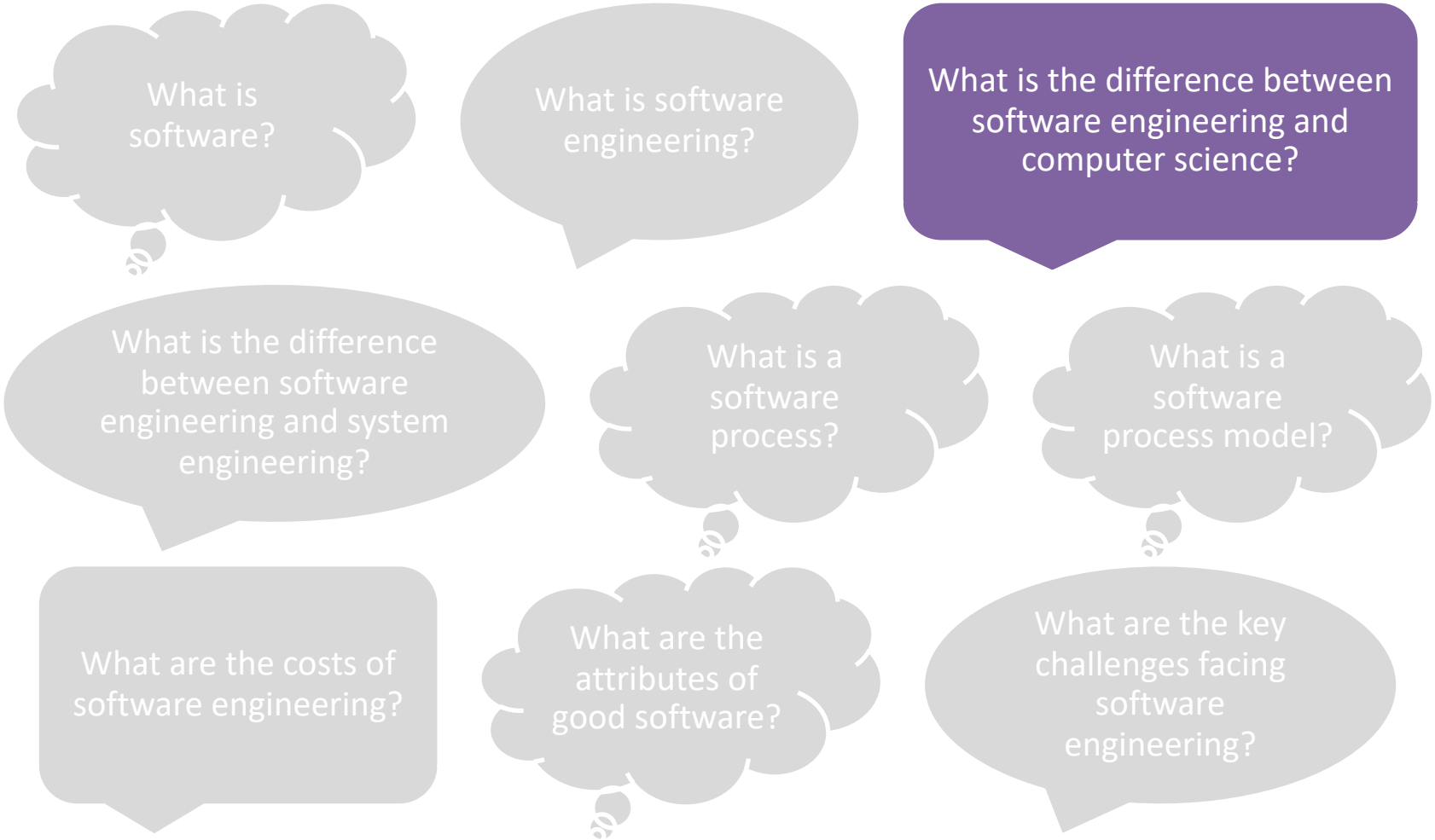
Software engineering

- Software engineering is an engineering discipline that is concerned with all aspects of software production from the early stages of system specification through to maintaining the system after it has gone into use.
- Engineering discipline
  - Using appropriate theories and methods to solve problems bearing in mind organizational and financial constraints.
- All aspects of software production
  - Not just technical process of development. Also project management and the development of tools, methods etc. to support software production.

Importance of software engineering

- More and more, individuals and society rely on advanced software systems. We need to be able to produce reliable and trustworthy systems economically and quickly.
- It is usually cheaper, in the long run, to use software engineering methods and techniques for software systems rather than just write the programs as if it was a personal programming project. For most types of system, the majority of costs are the costs of changing the software after it has gone into use.

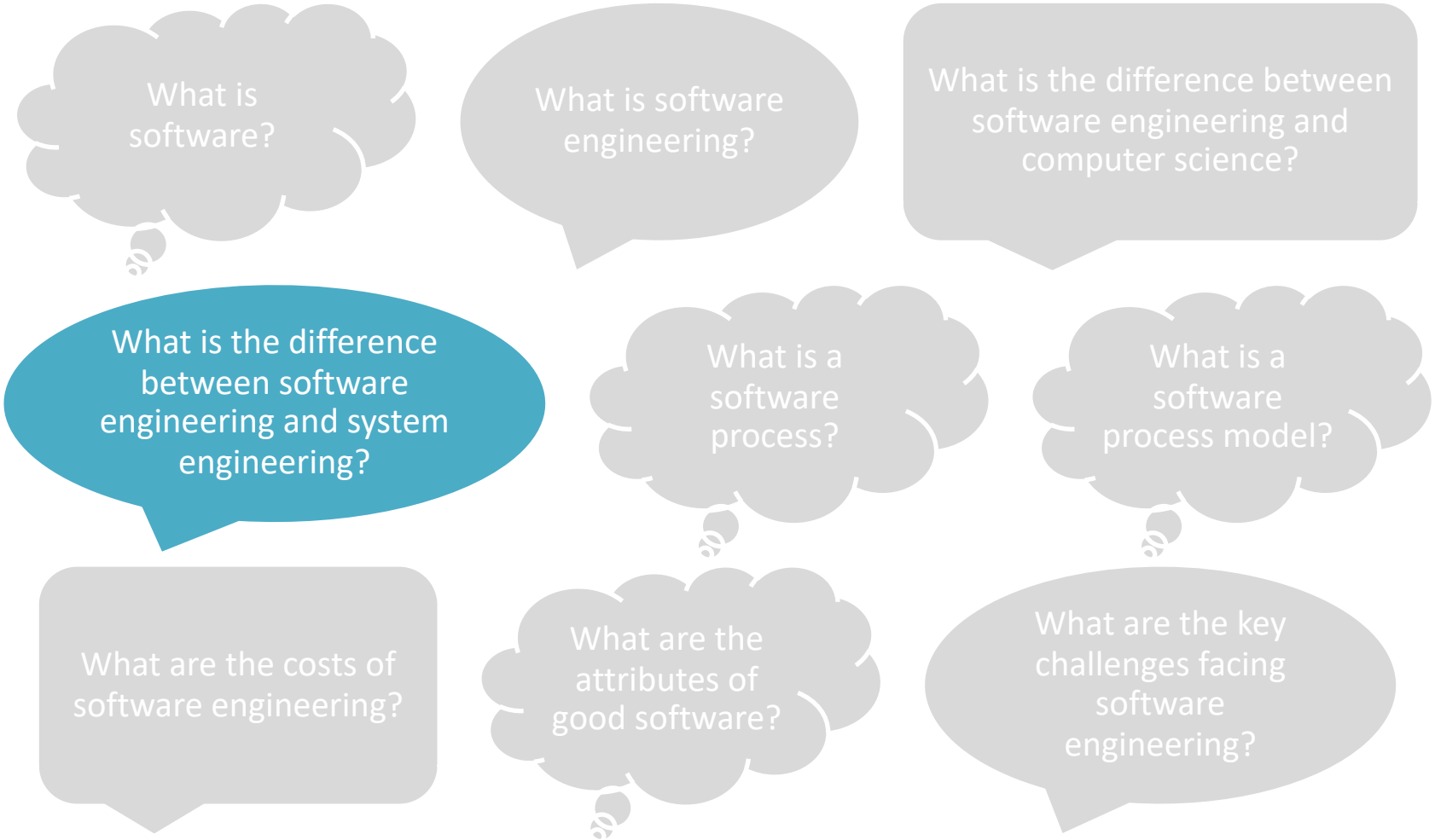
Frequently asked questions about software engineering



Software Engineering vs Computer Science

- Computer science focuses on theory and fundamentals; software engineering is concerned with the practicalities of developing and delivering useful software.

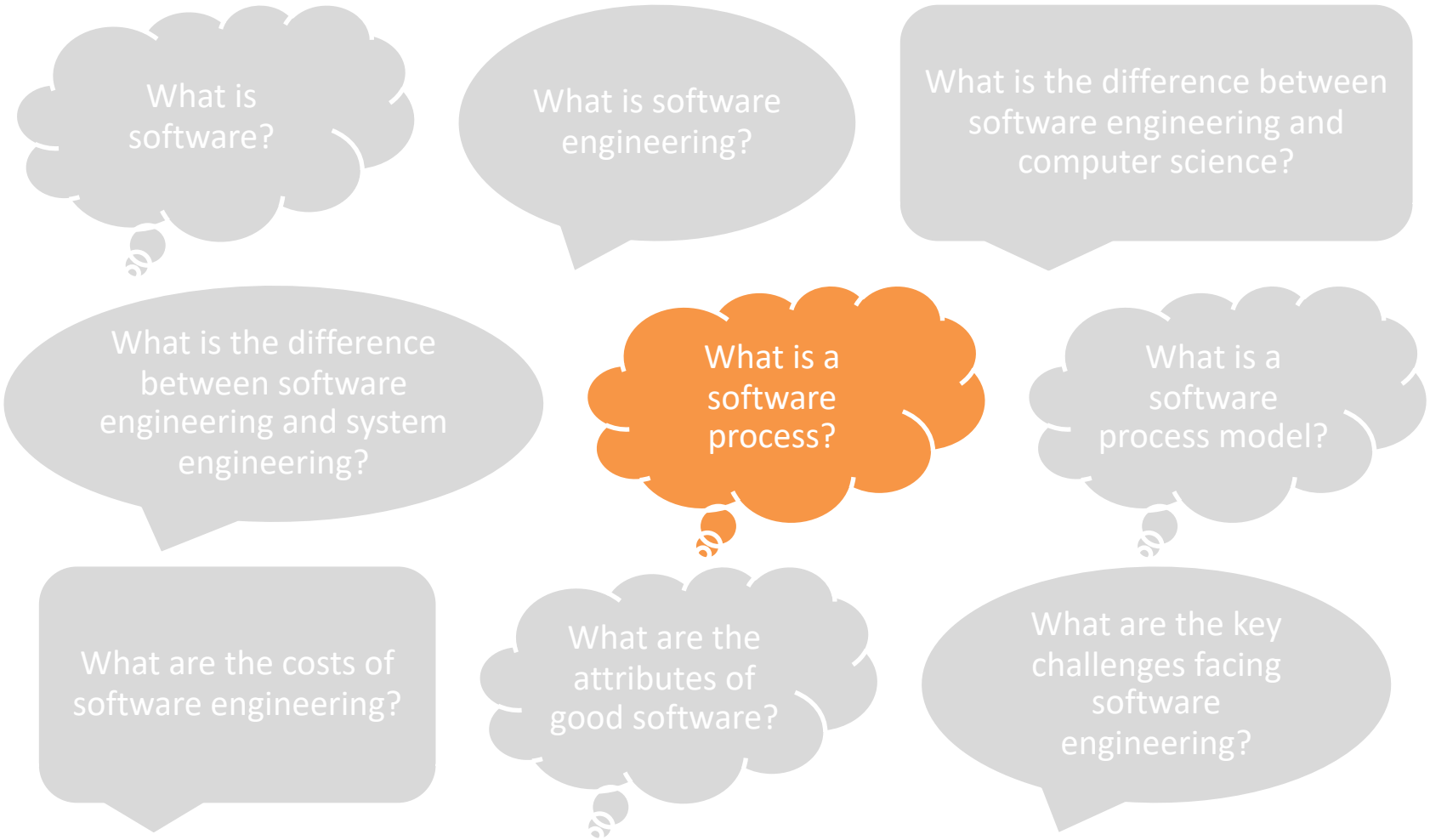
Frequently asked questions about software engineering



Software Engineering vs System Engineering

- System engineering is concerned with all aspects of computer-based systems development including hardware, software and process engineering.
- Software engineering is part of this process.
- System engineers are involved in system specification, architectural design, integration, and deployment.

Frequently asked questions about software engineering



Software process

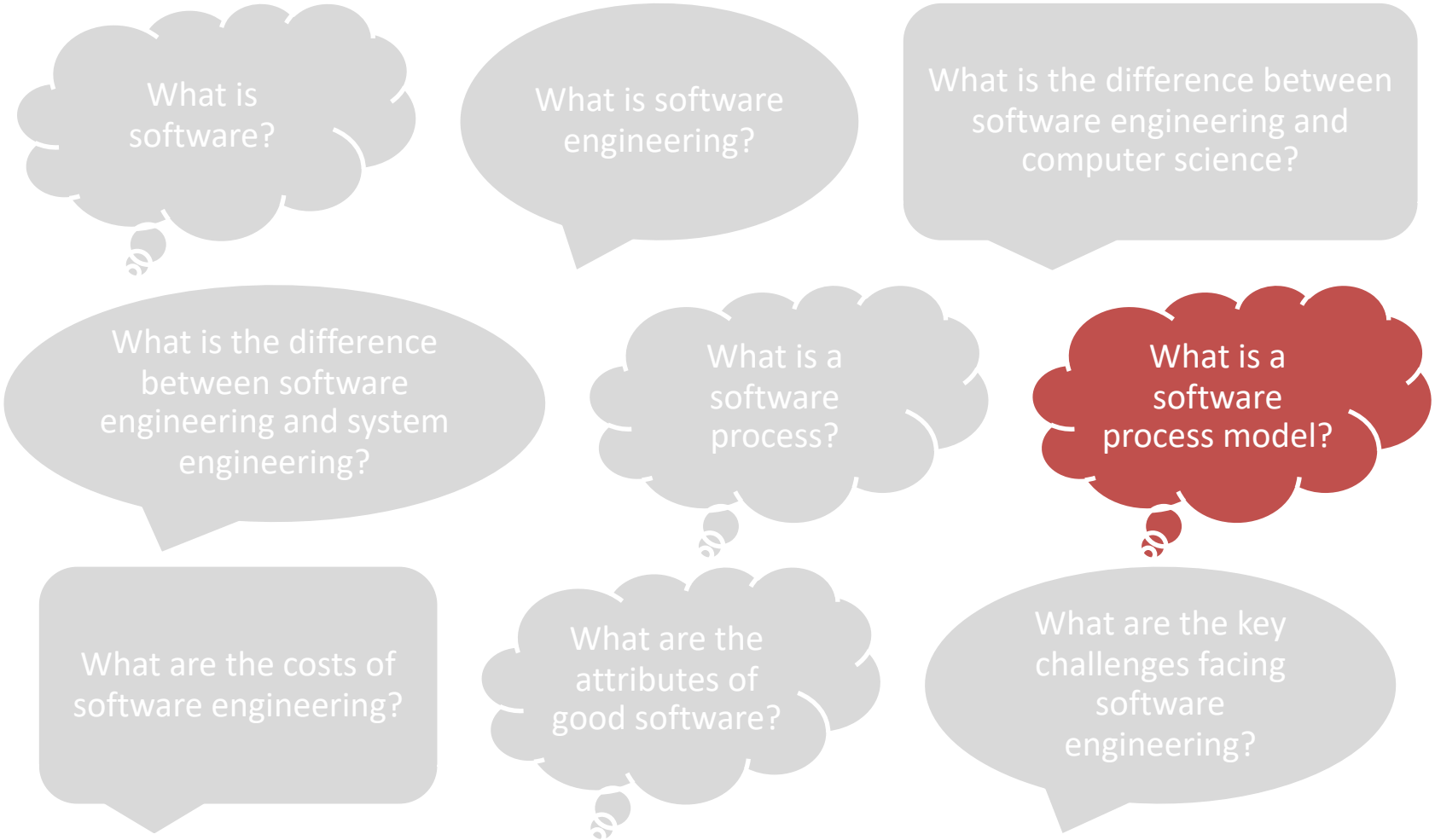
- Software engineers write sentences on the computers' language.
- However, this language is not appropriate to describe the problem that needs to be solved.
- Usually, customers describe their needs using a more ambiguous language, natural language, and software engineers need to understand their requirements and to write a solution.
- Therefore, the software engineer has to understand a problem, define how it can be solved by a computer, and write the commands that tell it how to solve the problem.
- This set of steps is known as the software process.

Software process activities

- A set of activities whose goal is the development or evolution of software.

- Software specification, where customers and engineers define the software that is to be produced and the constraints on its operation.
- Software development, where the software is designed and programmed.
- Software validation, where the software is checked to ensure that it is what the customer requires.
- Software evolution, where the software is modified to reflect changing customer and market requirements.

Frequently asked questions about software engineering



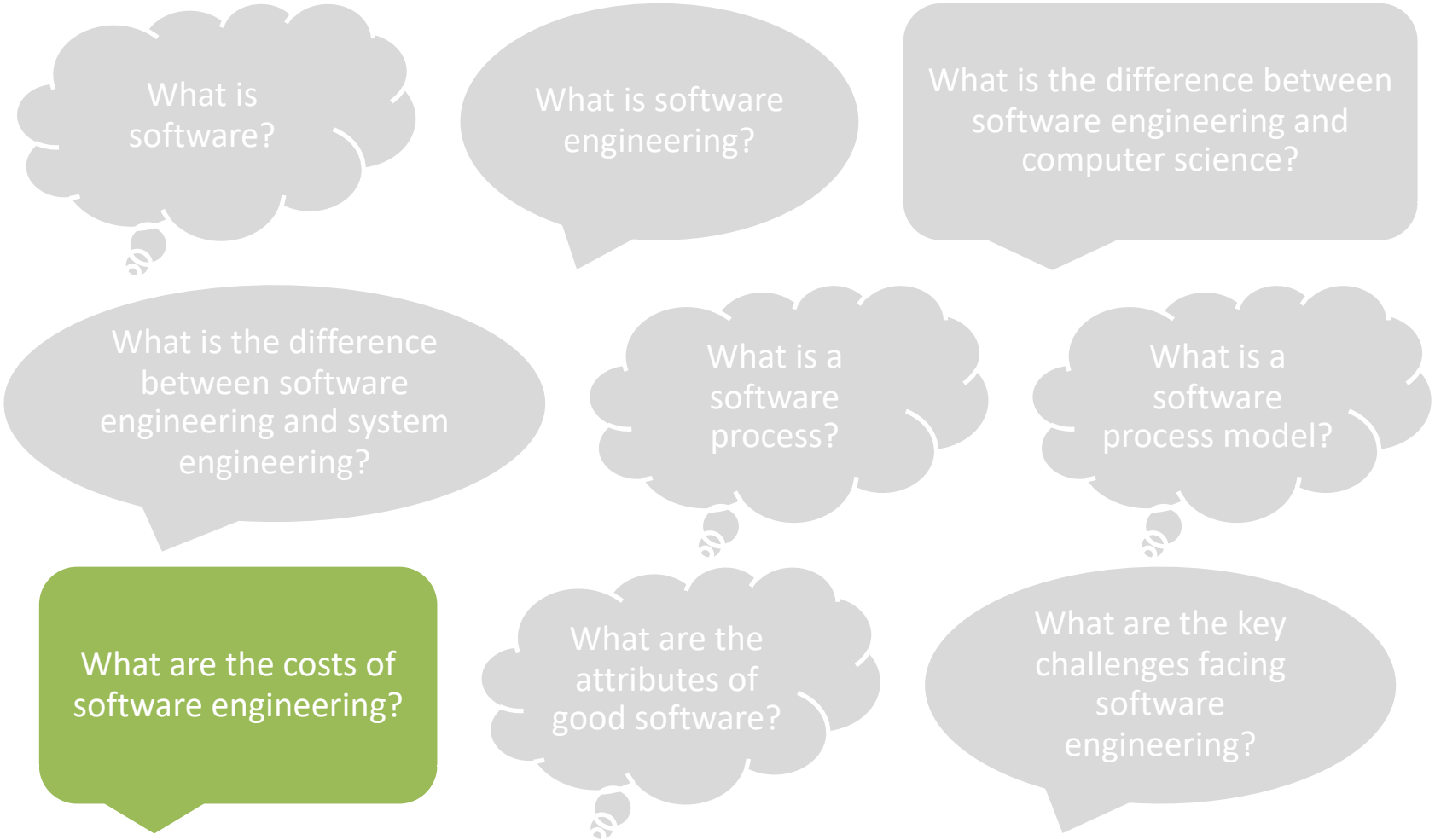
Software process model

- A simplified representation of a software process, presented from a specific perspective.

- Examples of process perspectives are:
  - Workflow perspective - sequence of activities.
  - Data-flow perspective - information flow.
  - Role/action perspective - who does what.

- Generic process models:
  - Waterfall.
  - Evolutionary development.
  - Formal transformation.
  - Integration from reusable components.

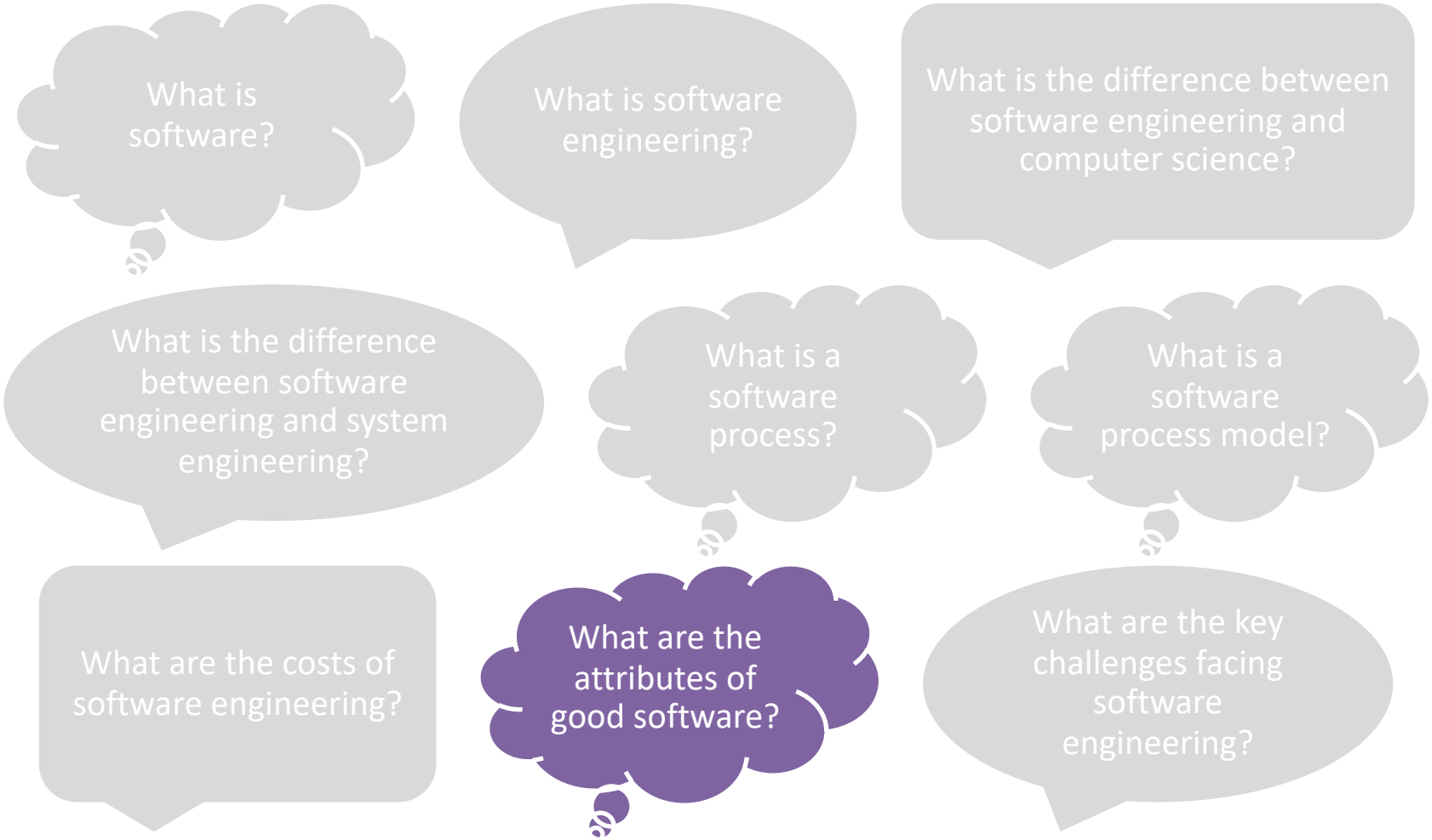
Frequently asked questions about software engineering



Costs of software engineering

- Roughly 50% of software costs are development costs, 50% are testing costs.
- For custom software, evolution costs often exceed development costs.
- Costs vary depending on the type of system being developed and the requirements of system attributes such as performance and system reliability.
- Distribution of costs depends on the development model that is used.

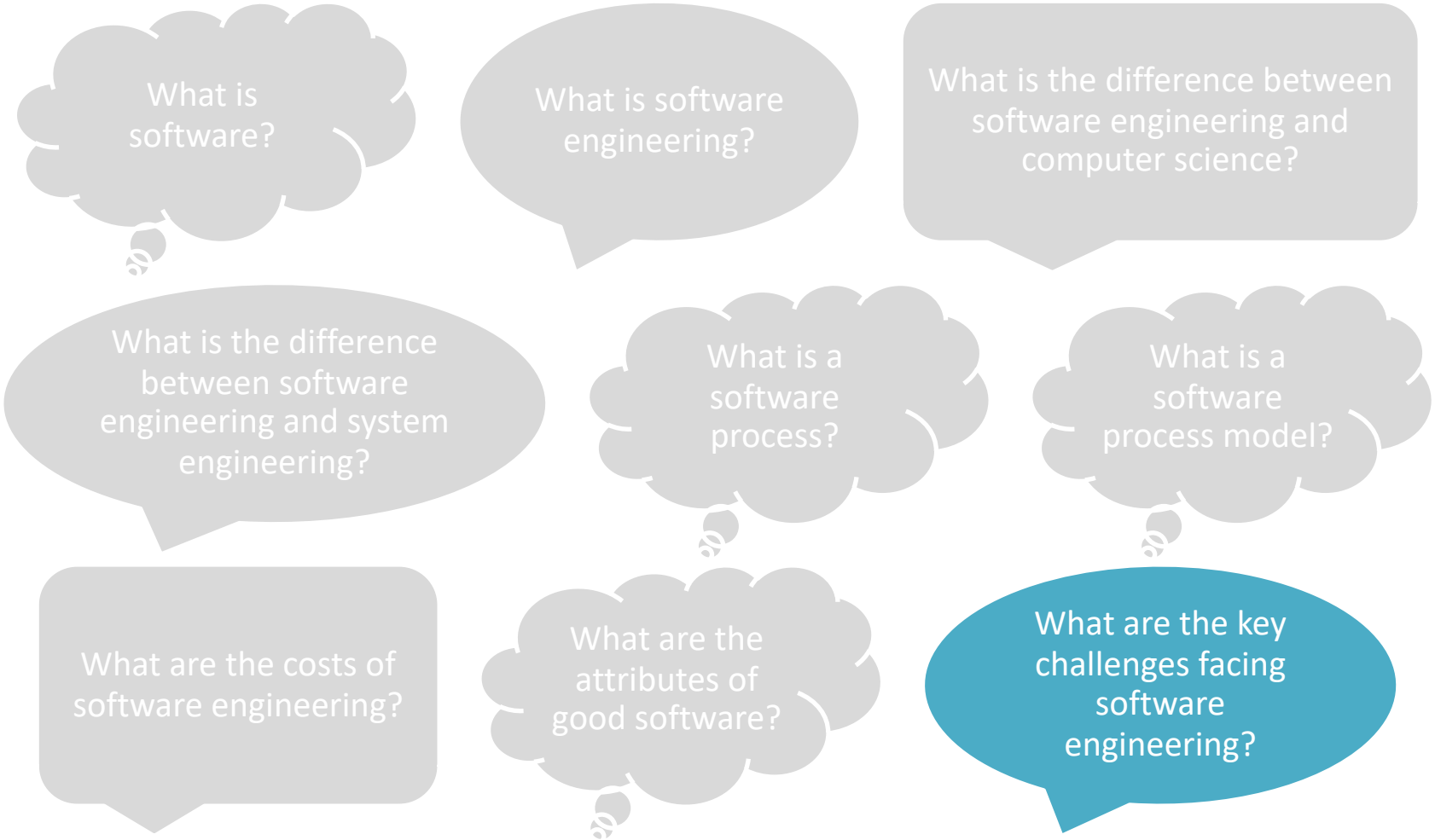
Frequently asked questions about software engineering



Attributes of good software

Product characteristic	Description
Maintainability	Software should be written in such a way so that it can evolve to meet the changing needs of customers. This is a critical attribute because software change is an inevitable requirement of a changing business environment.
Dependability and security	Software dependability includes a range of characteristics including reliability, security and safety. Dependable software should not cause physical or economic damage in the event of system failure. Malicious users should not be able to access or damage the system.
Efficiency	Software should not make wasteful use of system resources such as memory and processor cycles. Efficiency therefore includes responsiveness, processing time, memory utilisation, etc.
Acceptability	Software must be acceptable to the type of users for which it is designed. This means that it must be understandable, usable and compatible with other systems that they use.

Frequently asked questions about software engineering



Challenges facing software engineering

- ✧ Heterogeneity
  - Increasingly, systems are required to operate as distributed systems across networks that include different types of computer and mobile devices.
- ✧ Business and social change
  - Business and society are changing incredibly quickly as emerging economies develop and new technologies become available. They need to be able to change their existing software and to rapidly develop new software.

Challenges facing software engineering

- Security and trust
  - As software is intertwined with all aspects of our lives, it is essential that we can trust that software.
- Scale
  - Software has to be developed across a very wide range of scales, from very small embedded systems in portable or wearable devices through to Internet-scale, cloud-based systems that serve a global community.

Challenges facing software engineering

- ✧ Legacy Systems
  - Old, valuable systems must be maintained and updated.
- ✧ Delivery
  - There is increasing pressure for faster delivery of software.

Software project failure

✧ *Increasing system complexity*

- As new software engineering techniques help us to build larger, more complex systems, the demands change. Systems have to be built and delivered more quickly; larger, even more complex systems are required; systems have to have new capabilities that were previously thought to be impossible.

✧ *Failure to use software engineering methods*

- It is fairly easy to write computer programs without using software engineering methods and techniques. Many companies have drifted into software development as their products and services have evolved. They do not use software engineering methods in their everyday work. Consequently, their software is often more expensive and less reliable than it should be.

Software engineering diversity

- There are many different types of software system and there is no universal set of software techniques that is applicable to all of these.
- The software engineering methods and tools used depend on the type of application being developed, the requirements of the customer and the background of the development team.

Software engineering fundamentals

- Some fundamental principles apply to all types of software system, irrespective of the development techniques used:
  - Systems should be developed using a managed and understood development process. Of course, different processes are used for different types of software.
  - Dependability and performance are important for all types of system.
  - Understanding and managing the software specification and requirements (what the software should do) are important.
  - Where appropriate, you should reuse software that has already been developed rather than write new software.


HOW
WHY
WHERE
WHEN
WHERE
WHERE
WHY
WHO
WHO
WHEN
WHAT
WHAT
WHY
WHAT
WHEN
HOW
WHAT
WHEN
WHO
WHERE
WHY
HOW