

Recommended Skills and Knowledge for Software Engineers

_ B class 3T

By

_200711429 박상욱

_200711439 송인하

_200711444 양동은

Contents

1. Introduction

2. Computer Science VS Software Engineering

3. Recommended Software Engineering Skills and Knowledge

this->1 Computing Theory

this->2 Software Practice

this->3 Engineering Economy

this->4 Customer and Business Environment



1. Introduction



1. Introduction

- + There is already **reasonable agreement** across the industry on what constitutes an appropriate set of Skills and knowledge for computer Science.

Ex)

“Criteria for Accrediting Programs in Computer Science” by CSAB
A Survey of curricula available on the World Wide Web

- + However, there is **little agreement** on what constitutes appropriate skills and knowledge for Software Engineering and the relationship between computer science and software engineering

>> **That’s why software-engineering-degreed graduates often have widely varying skills and knowledge And this makes it difficult to assess the real value of a software degree.**

1. Introduction

+ What we need to know in this presentation...

- ^ What is the difference between Software Engineering and Computer Science?
- ^ What kinds of recommended Software Engineering Skills and Knowledge do we know?

2. Computer Science VS Software Engineering



2. Computer Science VS Software Engineering

+ A general definition of “Science” & “Engineering” by the basic vocabulary

Science

→ The pursuit of knowledge

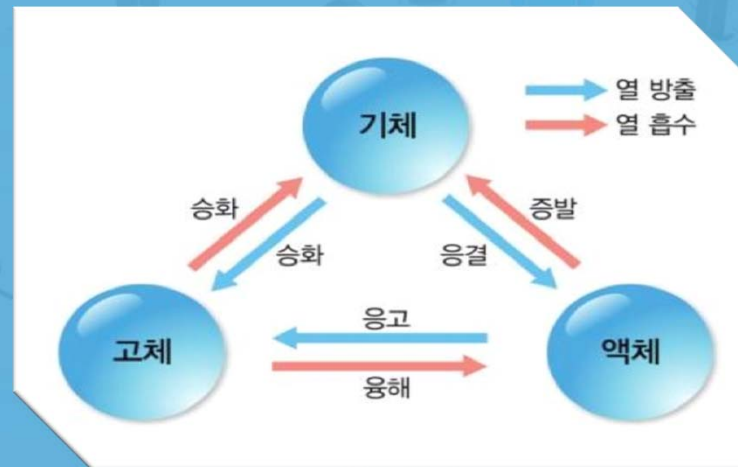
Engineering

→ The application of knowledge for the benefit of people

2. Computer Science VS Software Engineering

+ Chemistry as a science :

To expand our knowledge of chemical processes in order to better understand and explain phenomena that can be observed in the universe.



2. Computer Science VS Software Engineering

+ Chemical Engineering :

To apply the knowledge derived from “chemical science” to filling human needs and pursuit the practical aspects of chemical processes with an understanding of (engineering) economy



2. Computer Science VS Software Engineering

+ A simplified description of the general relationship
between science and engineering

$$\text{Engineering} = \text{Scientific theory} + \text{Practice} + \text{(Engineering) Economy}$$


2. Computer Science VS Software Engineering

+ A general definition of “computer science” & “software engineering” by the basic vocabulary



Computer science

A department of systematized knowledge **about computing** as an object of study; a system of knowledge covering general truths or the operation of general laws **of computing** especially as obtained and tested through scientific method



Software engineering

The profession in which a knowledge of the mathematical and **computing** sciences gained by study, experience, and practice is applied with judgement to develop ways to utilize, economically, **computing systems** for the benefit of mankind

2. Computer Science VS Software Engineering

+ A simplified description of the general relationship
between computer science and software engineering

Software Engineering = computing theory + Practice + (Engineering) Economy

2. Conclusion

+ What is the difference between Software Engineering and Computer Science?

- ^ Both computer science and software engineering deal with computers, computing, and software in common.
- ^ Computer Science is concerned with those things as a system of knowledge, together with the expansion of that knowledge
- ^ Software engineering is concerned with the application of those things to practical purposes, design, construction, and operation of reasonable computing systems

3. Recommended Software Engineering Skills and Knowledge



3. Recommended Software Engineering Skills and Knowledge

+ These recommended skills and knowledge are believed to

enable the proficient design, construction, and maintenance of cost-effective computing systems.

&

characterize “proper professional practice” for software engineers.

3. Recommended Software Engineering Skills and Knowledge

+ Keep in mind!

- ^ This recommendation describes a vision of “ideal” software engineer.
- ^ Individuals should be expected to have at least broad, but possibly shallow, proficiency across many of the skill and knowledge kernels and much more detailed proficiency in one specific areas of interest to them.
- ^ Rather than saying that every software engineer must be proficient in all of the skills or knowledge items listed here, the intent is that at least one software engineer on each software project ought to be proficient in each relevant skill.

3. Recommended Software Engineering Skills and Knowledge

+ A dictionary definition of Skill and Knowledge

A blue circular icon with a 3D effect and a white border, containing the word "Skill" in white text.

Skill

A learned power of doing something competently; a developed aptitude or ability

A blue circular icon with a 3D effect and a white border, containing the word "Knowledge" in white text.

Knowledge

Facts or ideas acquired by study, investigation, observation, or experience

3.1 Computing Theory

+ Knowledge of computing theory allows software engineers to:

- Propose a larger number of diverse designs than would otherwise be possible
- Identify and discard proposed designs that could not work (because they violate some known theory) earlier than otherwise possible

So

It can be evaluated on an economic basis to arrive at cost-effective designs.

3.1 Computing Theory

+ Table.

Recommended **computing theory** skills and knowledge

| | |
|--|----------------------------------|
| + Programming language concepts | + Data structure concepts |
| + Database system concepts | + Relational algebra |
| + Operating system concepts | + Software architectures |
| + Computer architectures | + Automata theory and Petri nets |
| + Computability theory and Turing machine theory | + Complexity theory |
| + Linguistics and parsing theory | + Computer graphics |
| + Set theory | + Predicate logic |
| + Formal proofs | + Induction |

3.2 Software Practice

- + The following table should be stated that software product engineering skills and knowledge apply as much to software maintenance as they do to software development

- + Table.

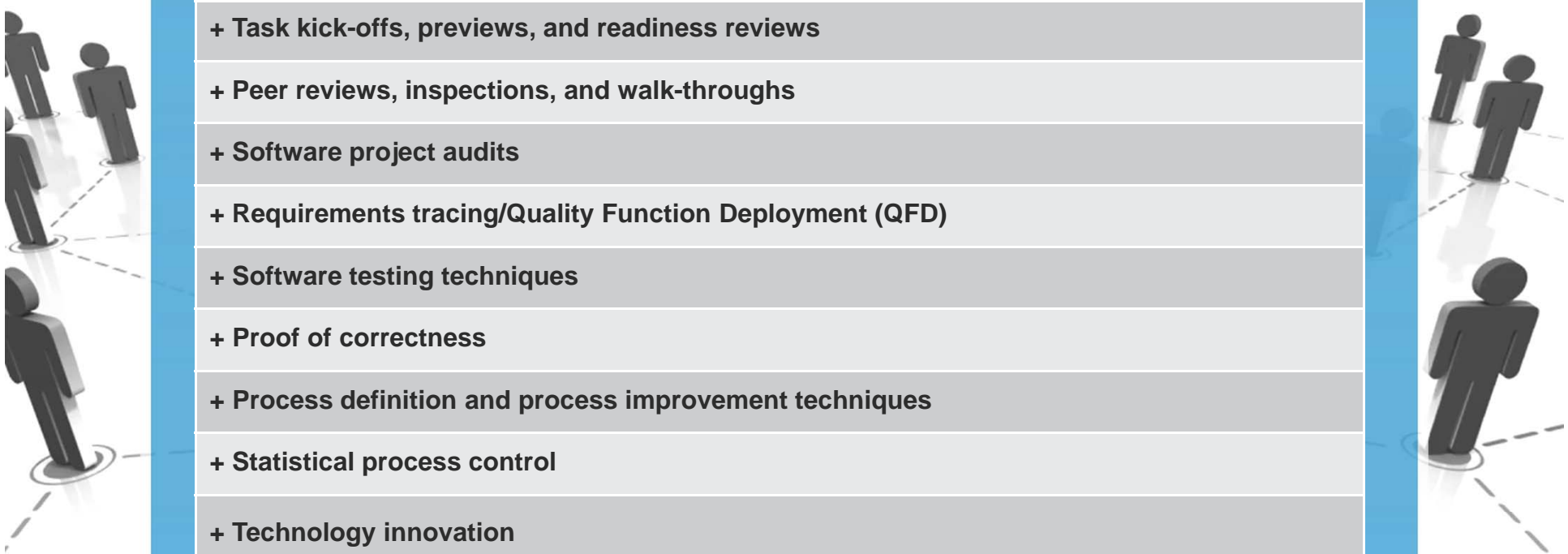
Recommended **software product engineering** skills and knowledge

| |
|--|
| + Requirements, analysis, and requirements engineering |
| + Software design |
| + Code optimization and semantics preserving transformations |
| + Human-computer interaction, and usability engineering |
| + Specific programming languages |
| + Debugging techniques |
| + Software-software and software-hardware integration |
| + Product family engineering techniques and reuse techniques |
| + CASE/CASE tools |

3.2 Software Practice

+ Table.

Recommended **software quality assurance(SQA)** skills and knowledge



| |
|--|
| + Task kick-offs, previews, and readiness reviews |
| + Peer reviews, inspections, and walk-throughs |
| + Software project audits |
| + Requirements tracing/Quality Function Deployment (QFD) |
| + Software testing techniques |
| + Proof of correctness |
| + Process definition and process improvement techniques |
| + Statistical process control |
| + Technology innovation |

3.2 Software Practice

- + The software product need to be packaged and delivered to the customer in a form that they can use.
- + The customer may also need help in using, maintaining, or repairing the product

+ Table.

Recommended **software product deployment** skills and knowledge

- | |
|--------------------------------------|
| + User documentaion techniques |
| + Product packaging techniques |
| + System conversion techniques |
| + Customer support techniques |
| + General technology transfer issues |

3.2 Software Practice

- + Software engineering management is intended to provide the needed coordination both within the software organization itself and between the software organization and its neighboring organizations.

+ Table.

Recommended **software engineering management** skills and knowledge

| | |
|--|----------------------------------|
| + Risk assessment and risk management | + Project planning |
| + Alternative software lifecycles | + Organizational structures |
| + Organizational behavior | + Project tracking and oversight |
| + Cost management, schedule management, and resource management | |
| + Metrics, goal-question-metric paradigm, and measurement theory | |
| + Configuration management and change | + Management |
| + Supplier and subcontract management | + Effective meeting skills |
| + Effective communication skills | + Negotiation skills |

3.3 Engineering Economy

Economy

Thrifty and efficient use of resources

Engineering
Economy

Ways to produce the best interest of the enterprise to invest its limited resources in a proposed technical endeavor

3.3 Engineering Economy

- + Engineering economy and its relevance to software engineering
- + Leon Levy said ...

Software economics has often been misconceived as the means of estimating the cost of programming projects.



But economics is primarily a science of choice, and software economics should provide methods and models for analyzing the choices that software projects must make.

3.3 Engineering Economy

+ Table.

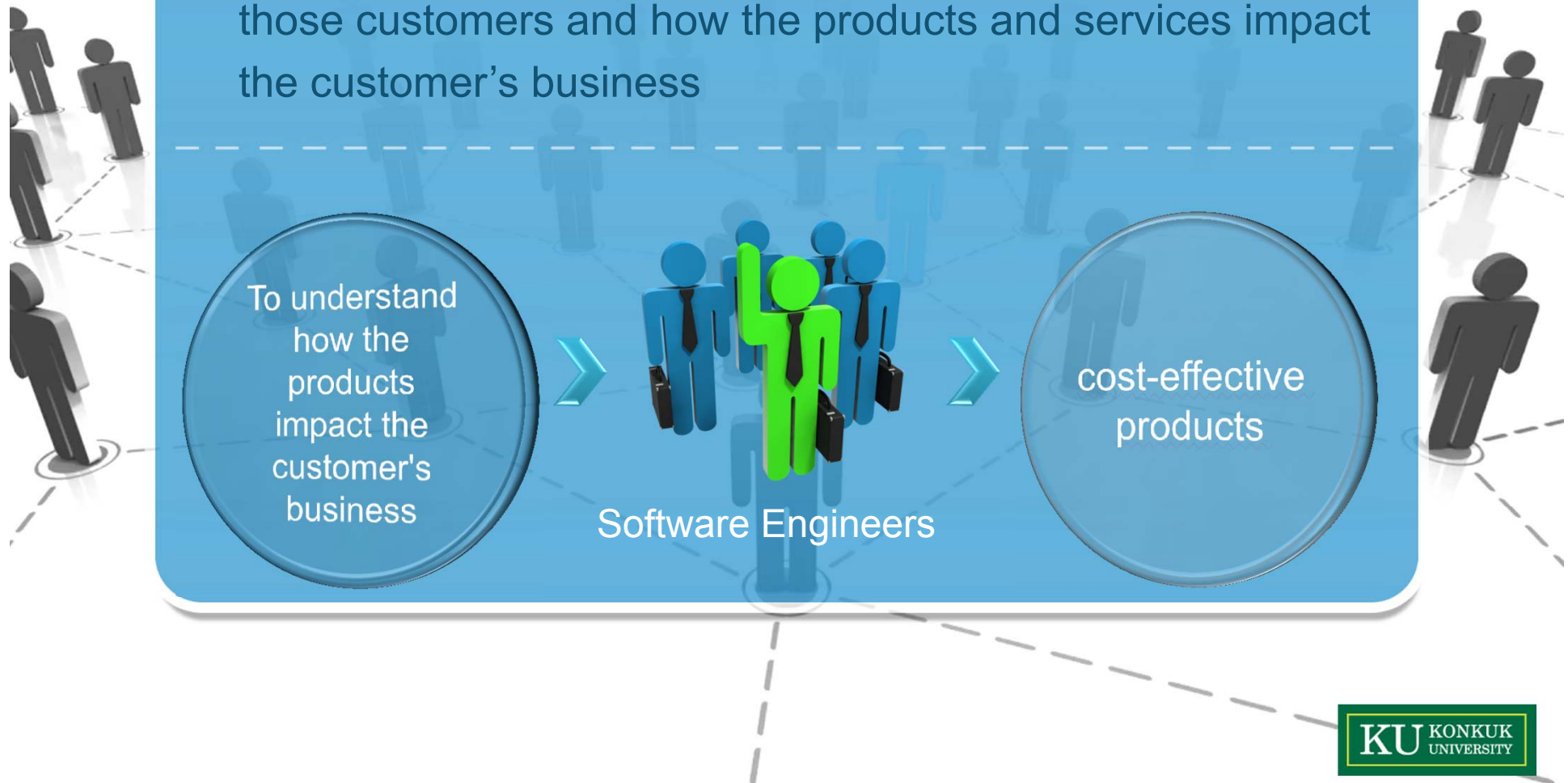
Recommended **engineering economy** skills and knowledge

| | |
|---------------------------------------|--|
| + Time value of money (interest) | + Economic equivalence |
| + Inflation | + Income taxes |
| + Decision making among alternatives | + Decision making under risk and Uncertainty |
| + Evaluating replacement alternatives | + Evaluating public activities |
| + Breakeven | + Optimization |



3.4 Customer and Business Environment

- + Soft engineers cannot offer products and services that will be cost-effective to customers without understanding both those customers and how the products and services impact the customer's business



3.4 Customer and Business Environment

+ Software engineers need the following knowledge:

- > Who is the customer and what is their business?
- > What do they use our products and services for?
- > When, where, and why are our products and services used?
- > Are our products and services being used in a way different than originally intended? If so, why?
- > How do our products and services affect the customer's business?
- > What external restrictions or regulations impact the ability to deliver products and services to customer(s)?



3.4 Customer and Business Environment

+ Table.

Recommended **customer and business environment** skills and knowledge

| |
|---|
| |
| + Customer satisfaction assessment techniques |
| + Competitive benchmarking techniques |
| + Technical communication |
| + Intellectual property law |
| + Ethics and professionalism |

3. Conclusion

+ What kinds of recommended Software Engineering Skills and Knowledge do we know?

+ Computing theory skills and knowledge

>> Programming language, Data structure and Database system concepts, etc.

+ Software product engineering skills and knowledge

>> Software design, Debugging techniques, CASE/CASE tools, etc.

+ Software quality assurance skills and knowledge

>> previews, inspections, Software project audits ,etc.

3. Conclusion

+ Software product deployment skills and knowledge

>> User documentaion, Product packaging and System conversion techniques, etc.

+ Software engineering management skills and knowledge

>> Risk, cost and schedule management, etc.

+ Engineering economy skills and knowledge

>> Time value of money, Income taxes, Optimization, etc.

+ Customer and business environment skills and knowledge

>> Intellectual property law, Customer satisfaction assessment, etc.

The End



_Thanks for your attentions!