Programming (coding)

specific task in SD, write computer instruction → solve given problem

Programmer —code→ software programs

program designs ← software developer

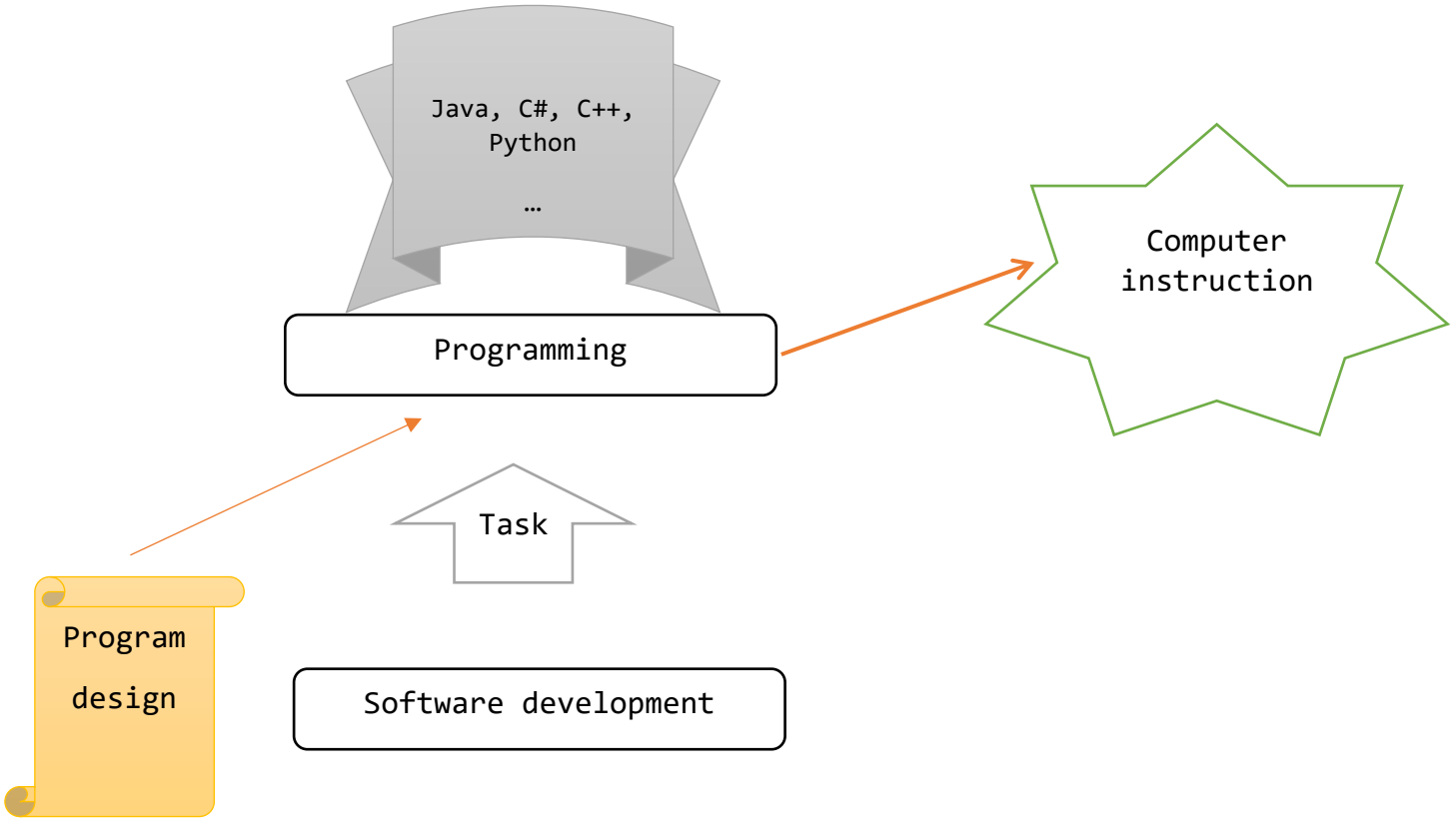
↓

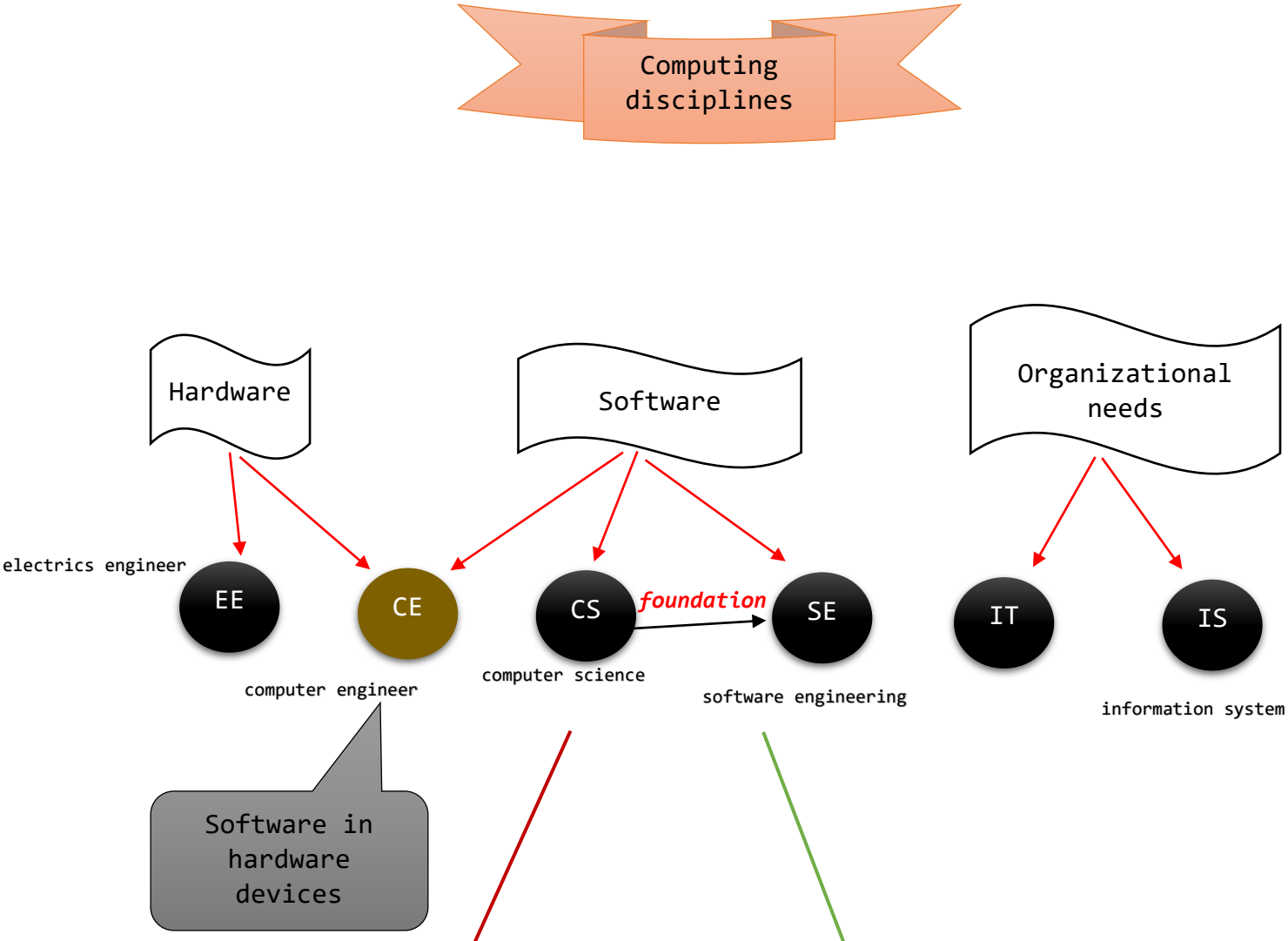
Computer instructions

Software development

broader creative process, construct a software → particular purpose

develop application, underlying systems




Computing disciplines
Hardware
Software
Organizational needs
electrics engineer
EE
CE
CS
foundation
SE
IT
IS
computer engineer
computer science
software engineering
information system
Software in hardware devices
software
robust real-world requirements
foundational concepts + techniques + tools
clever software
new alogrithm
language
express computing ideas
development process
complex software

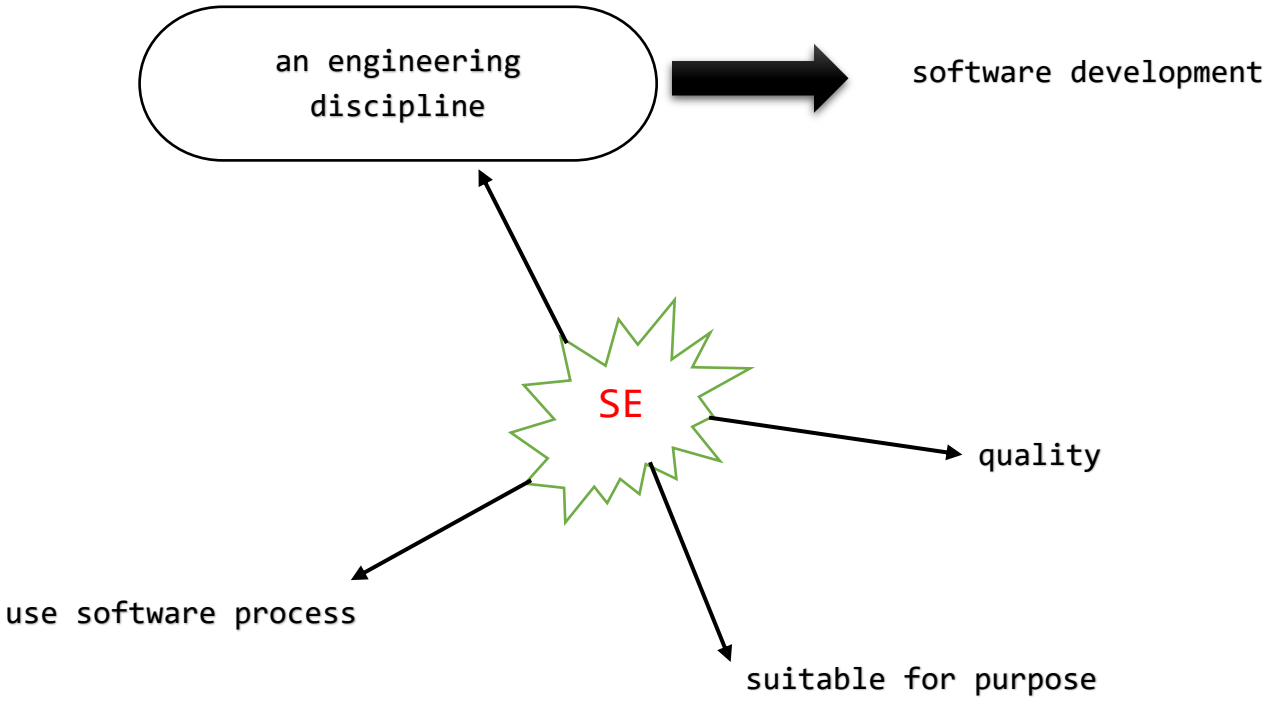
SE & Enginerring:

- decision-based
- measure things
- disciplined process
- … team
- multiples roles
- systematically
- resuse + design
- advance priciples + standard
- best practices

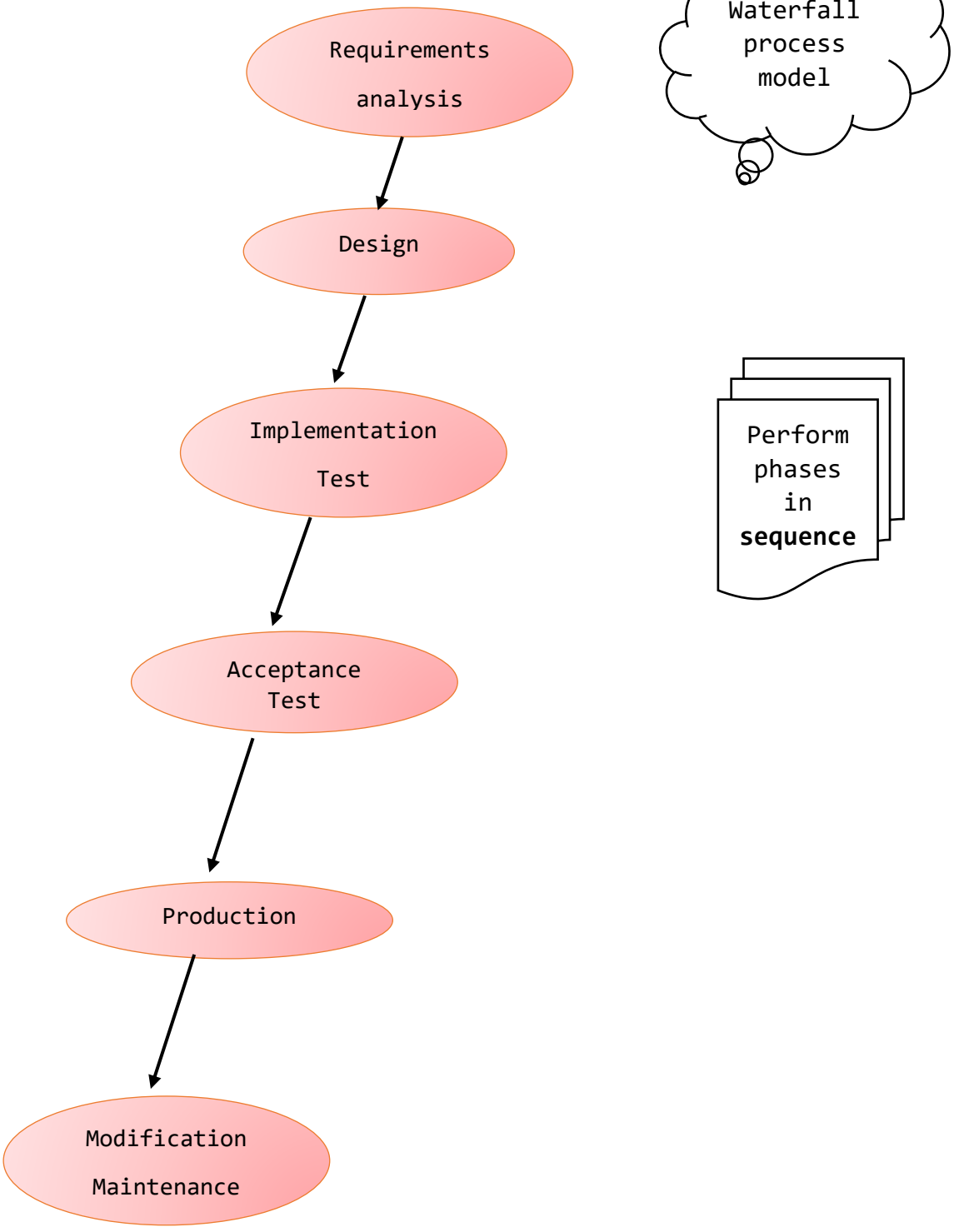
SE	*Engineering*
foundation: CS	natural sciences
discrete mathematics	continuous mathematics
(abstract) *logical* entities	(concrete) physical artifacts
no "manufacturing"	manufacturing phase
"maintenance": *evolution*	conventional wear & tear

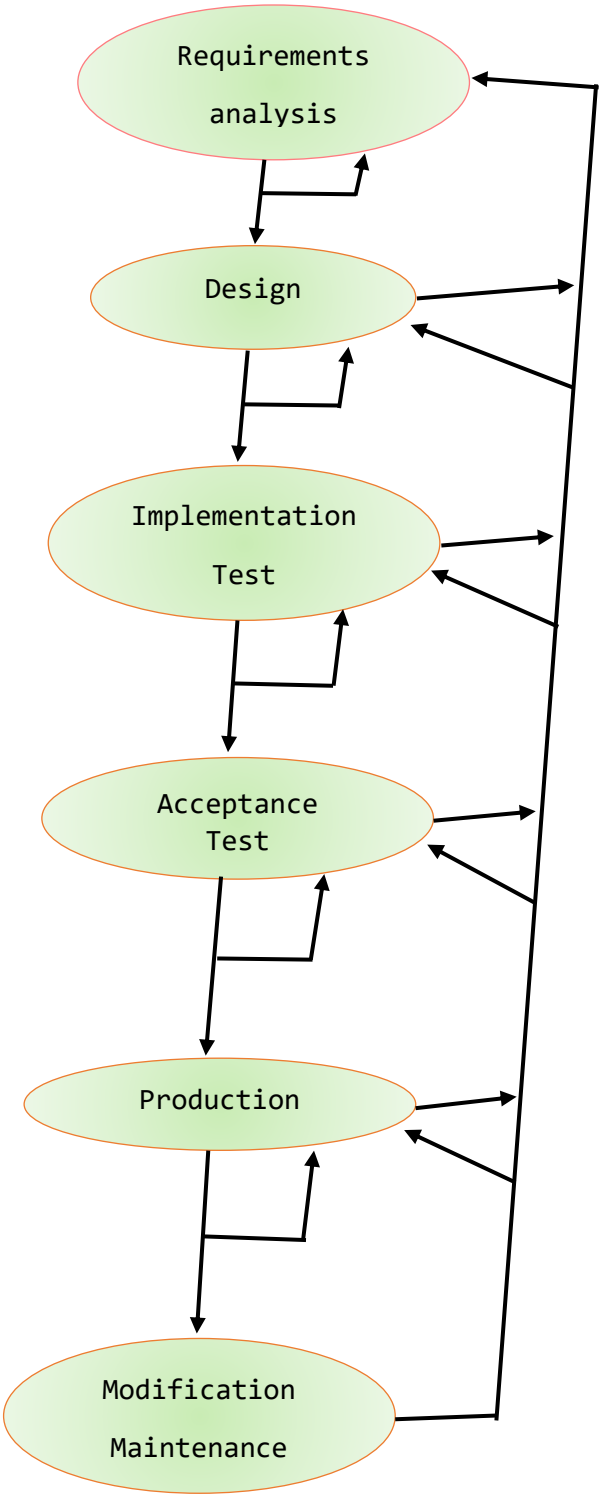
Software: non-physical

interact → other real-world systems (social, physical)


an engineering discipline
software development
SE
quality
use software process
suitable for purpose

Software development life cycle

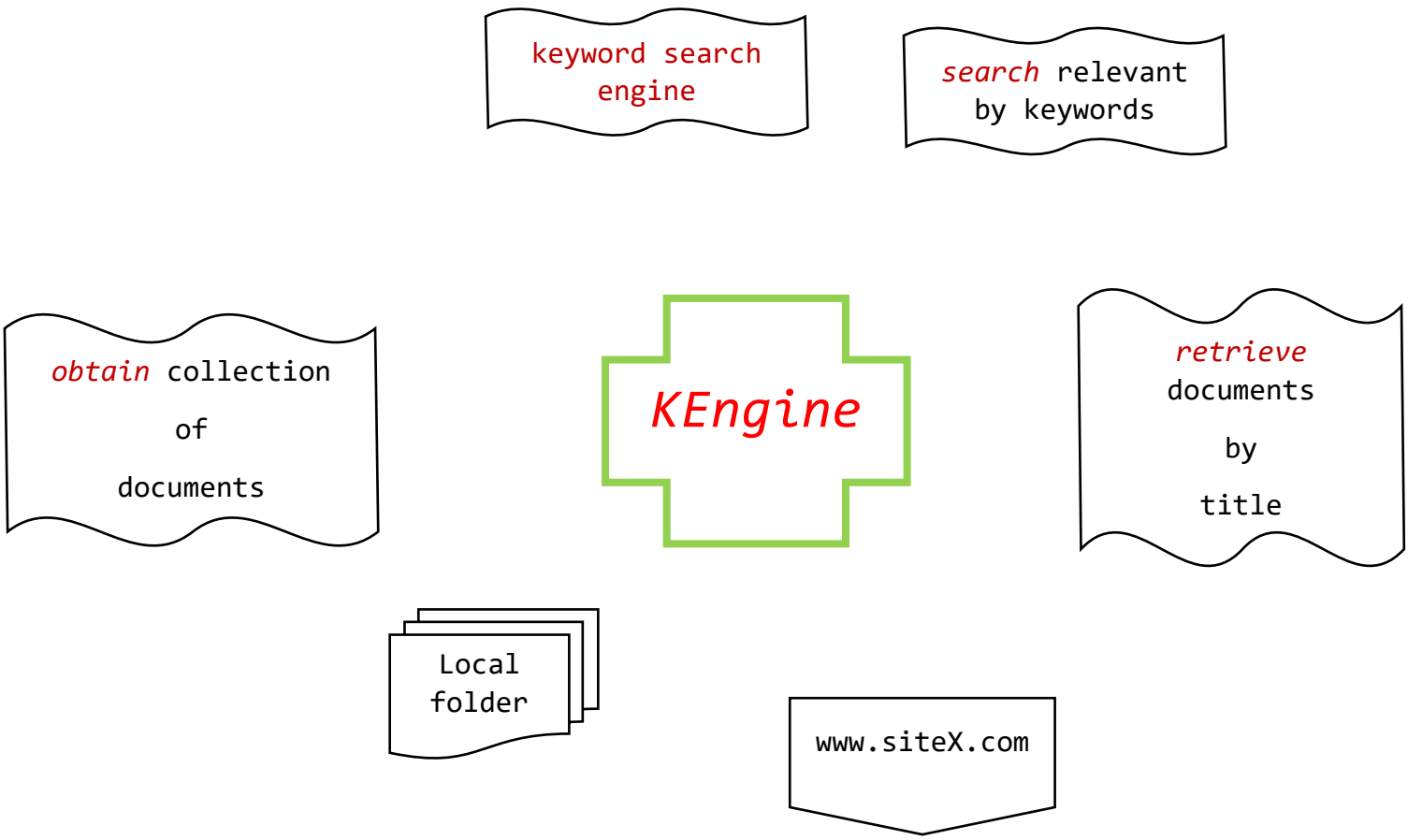




Spiral
model

perform phases
in **iterations**
with
feedback loops

Case study



Document: sequence of words

html document: title + body

Word: provide data (~~html tags~~)

keyword non-keyword

Document *d1*

```
<html>
<head>
      <title> welcome to my page </title>
</head>
<body>
<p>
this is a test page to test the simple Doc parser
</body>
</html>
```

Document *d2*

```
<html>
<head>
      <title> welcome to my page </title>
</head>
<body>
<p>
another test page
</body>
</html>
```

keyword frequencies

<”test”, <d1, 2>> <”test”, <d2, 1>>

<”page”, <d1, 1>> <”page”, <d2, 1>>

<”to”, <d1, 1>>

<”simple”, <d1, 1>>

<”Doc”, <d1, 1>>

<”parser”, <d1, 1>>

<”another”, <d2, 1>>

Query: a set of keywords → merge it → team

result: matches are sorted in **descending** order

Query: {“test”}

➔ result: { <d1, 2>, <d2, 1> }

Query: {“test”, “page”}

➔ result: { <d1, 3>, <d2, 2> }

Query: {“simple”, “Doc”, “parser”}

➔ result: { <d1, 3> }