投票 最多可选1项

设置

Is everything ready? (device, Internet and yourself)

A Yes

B No

提交

Software Engineering

Part 1
Software Process

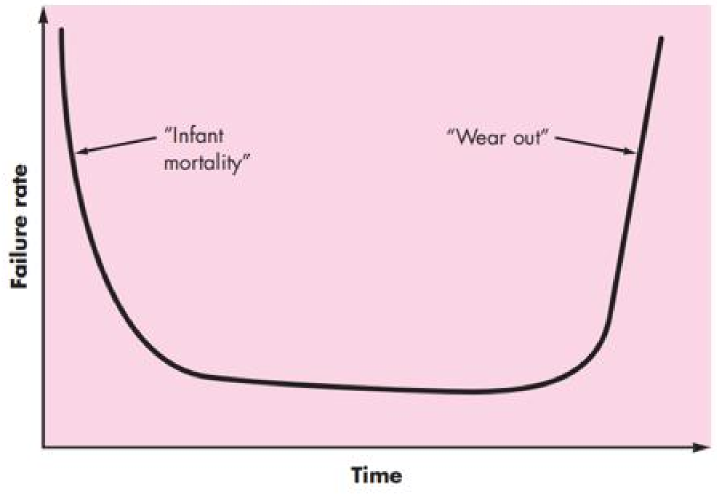
Chapter 3
Software Process Structure

Reproduced by Ning Li , 2022

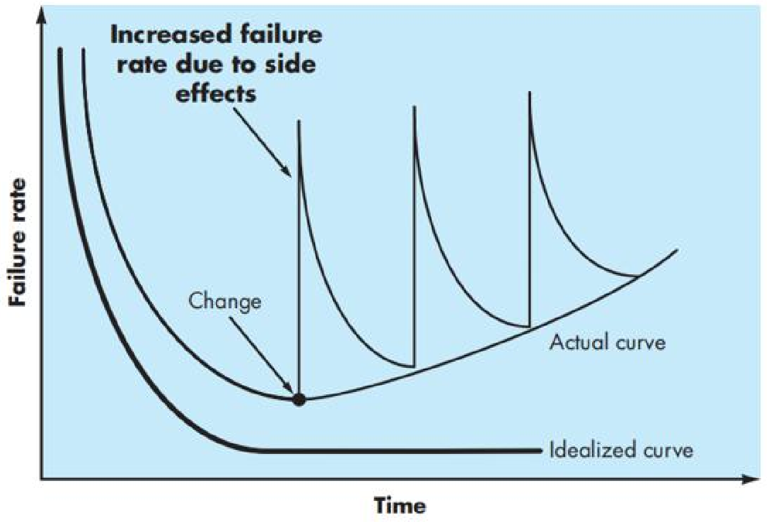
Review - 1

- What is software
  - instructions + data + documentation
  - nature?
  - difference with hardware

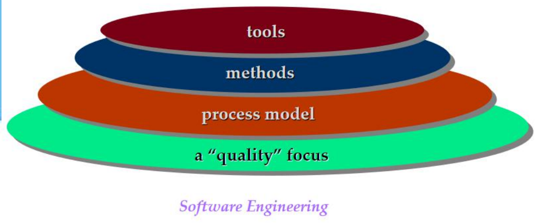
    (wear out, complex maintenance)



bathtub curve

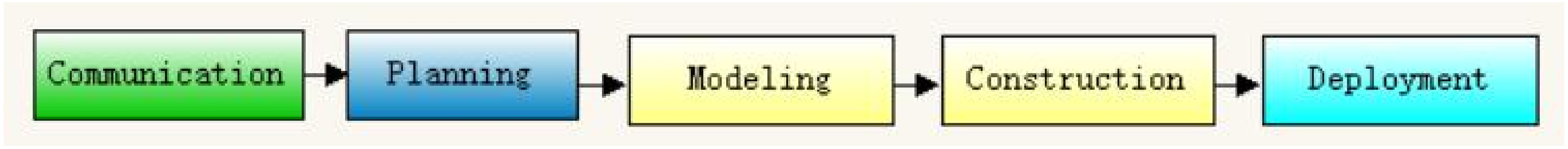


Review - 2



- What is software Engineering
  - application of engineering to software + research
  - layered technology (quality -> process ->method -> tool)
  - process framework
    - framework activities

      Communication, Planning, Modeling, Construction, Deployment
    - umbrella activities
  - practice: understand , plan , carry out, examine



Question

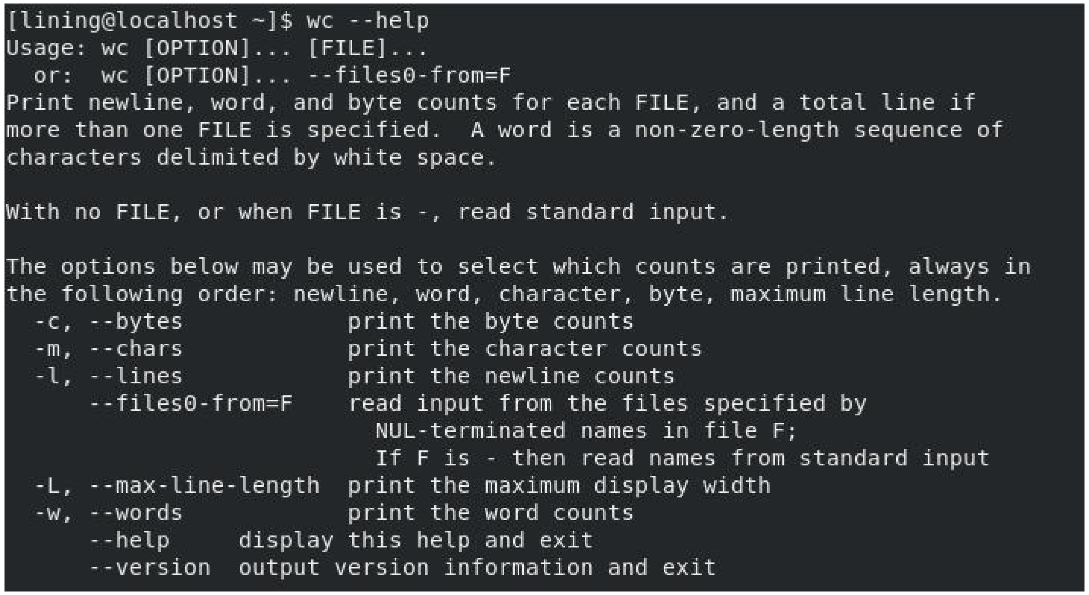
- As a computer student, what kinds of useful software can you want/try to develop to support the coronavirus outbreak?

  Please think the real requirements!

coronavirus

Practice

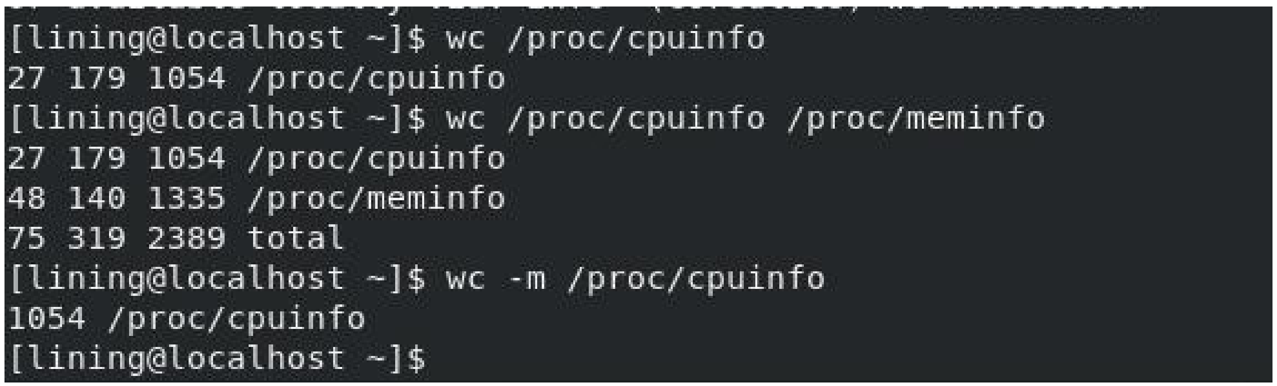
Design and implement a software like wc command in Linux



```
[lining@localhost ~]$ wc --help
Usage: wc [OPTION]... [FILE]...
  or:  wc [OPTION]... --files0-from=F
Print newline, word, and byte counts for each FILE, and a total line if
more than one FILE is specified.  A word is a non-zero-length sequence of
characters delimited by white space.

With no FILE, or when FILE is -, read standard input.

The options below may be used to select which counts are printed, always in
the following order: newline, word, character, byte, maximum line length.
  -c, --bytes            print the byte counts
  -m, --chars            print the character counts
  -l, --lines            print the newline counts
      --files0-from=F    read input from the files specified by
                           NUL-terminated names in file F;
                           If F is - then read names from standard input
  -L, --max-line-length  print the maximum display width
  -w, --words            print the word counts
      --help     display this help and exit
      --version  output version information and exit
```



```
[lining@localhost ~]$ wc /proc/cpuinfo
27 179 1054 /proc/cpuinfo
[lining@localhost ~]$ wc /proc/cpuinfo /proc/meminfo
27 179 1054 /proc/cpuinfo
48 140 1335 /proc/meminfo
75 319 2389 total
[lining@localhost ~]$ wc -m /proc/cpuinfo
1054 /proc/cpuinfo
[lining@localhost ~]$
```

Software Process Structure

3.1 A Generic Process Model

3.2 Defining a Framework Activity

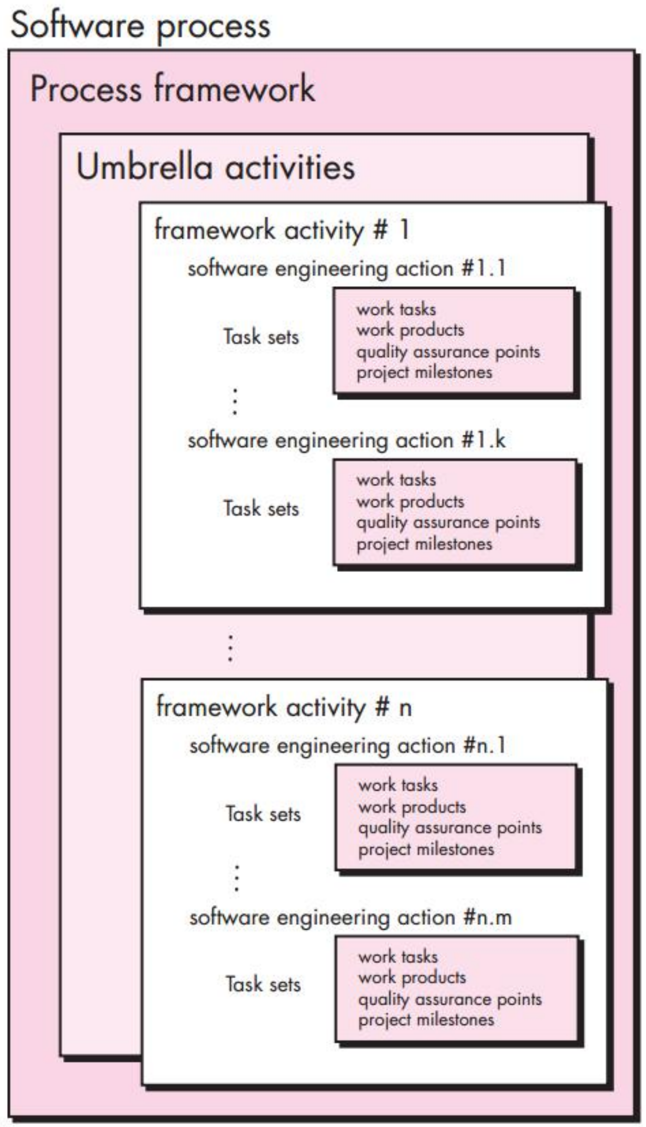
3.3 Identifying a Task Set

3.4 Process Patterns

3.5 Process Assessment and Improvement

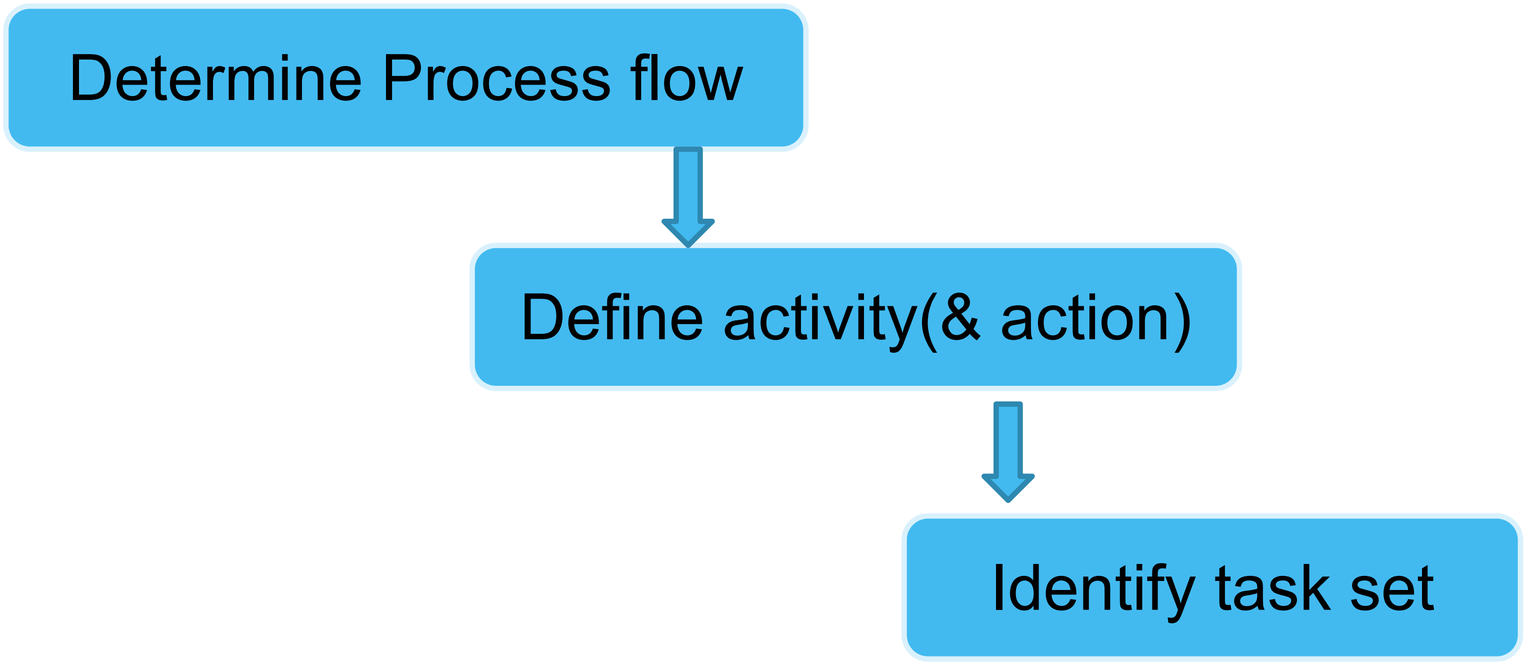
3.1 A Generic Process Model

Process
- activity1
  - action 1.1
    - task 1.1.1
    - ...
  - action1.2

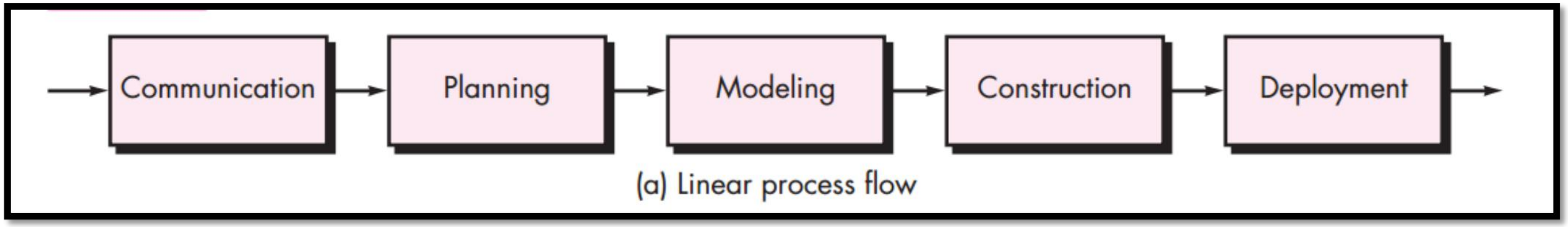


3.1 A Generic Process Model

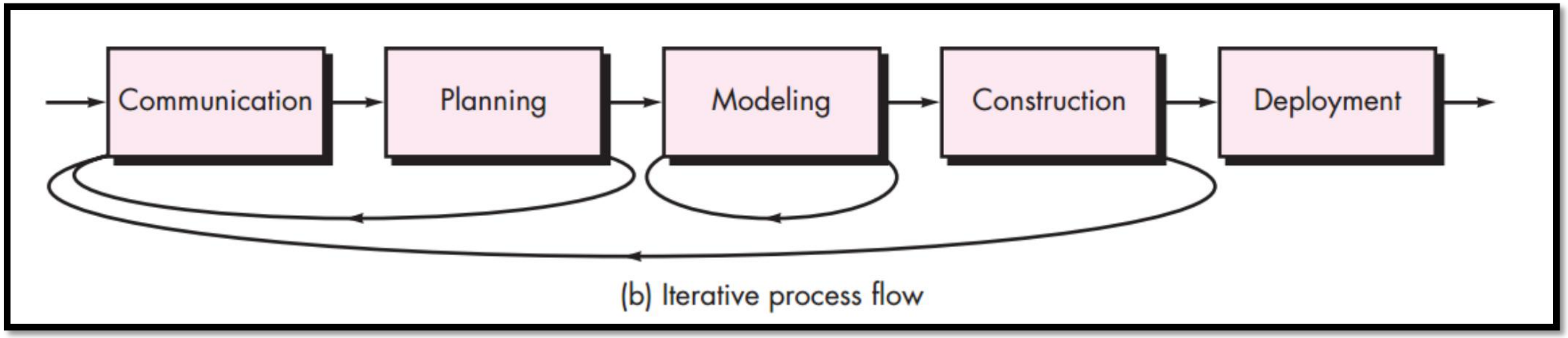
How to determine Software Process



3.1 Process Flow

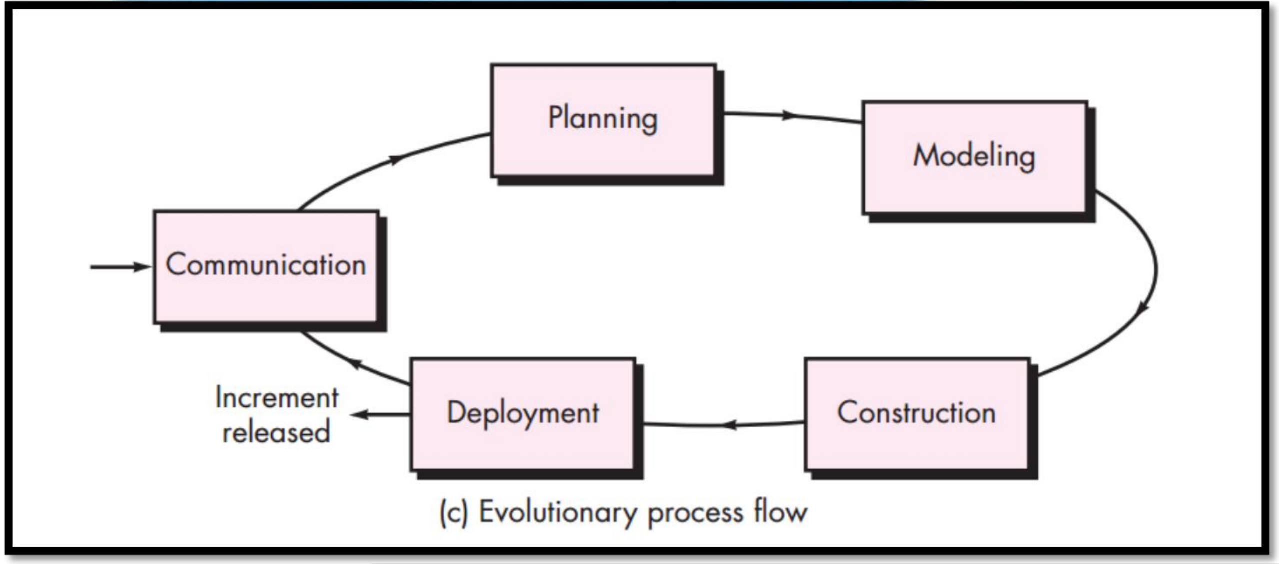


(a) Linear process flow

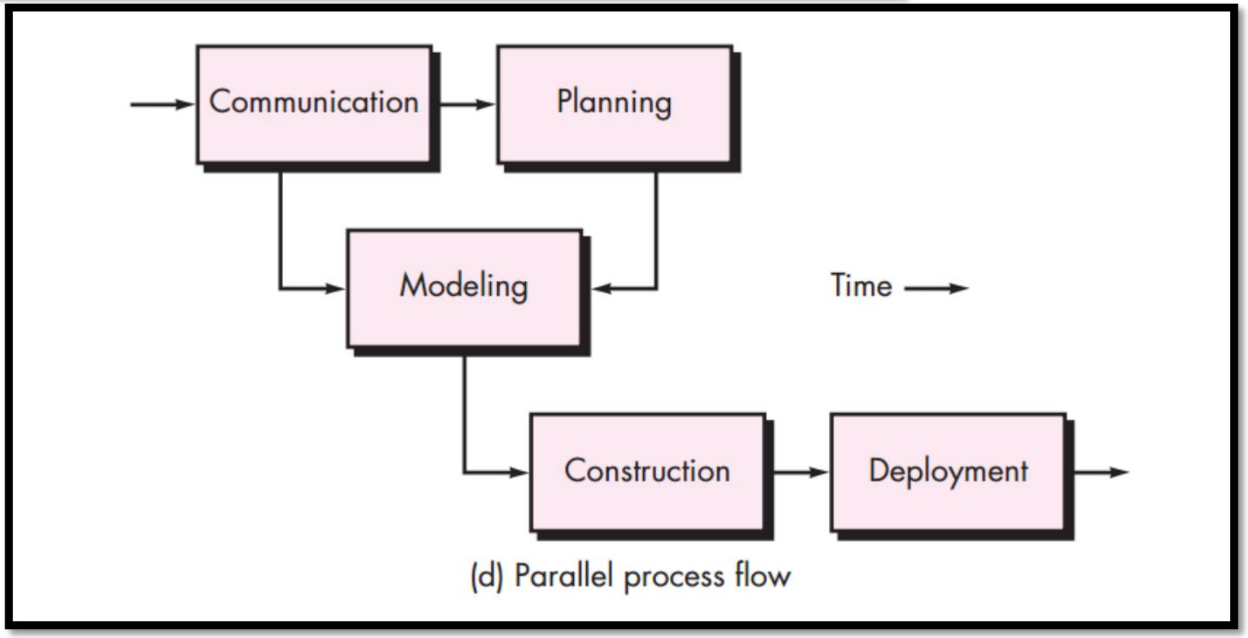


(b) Iterative process flow

3.1 Process Flow



(c) Evolutionary process flow



(d) Parallel process flow

单选题 1分

设置

此题未设置答案，请点击右侧设置按钮

I think,all the options are correct ,I just choose C

Which statement about process flow is NOT right?

A A linear process flow executes each of the framework activities in sequence.

B An iterative process flow repeats one or more of the activities before proceeding to the next.

C Each circuit can execute part of the five activities in evolutionary process flow.

D Parallel process flow executes one or more activities in parallel with other activities.

https://softwareengineeringmca.blogspot.com/2016/10/a-generic-process-model-iterative-process-flow-linear-process-evolutionary-process-parallel-process.html

提交

3.2 Defining a Framework Activity

Example:

For a small software project requested by one person (at a remote location) with simple, straightforward requirements. We can define the **communication activity** as following:

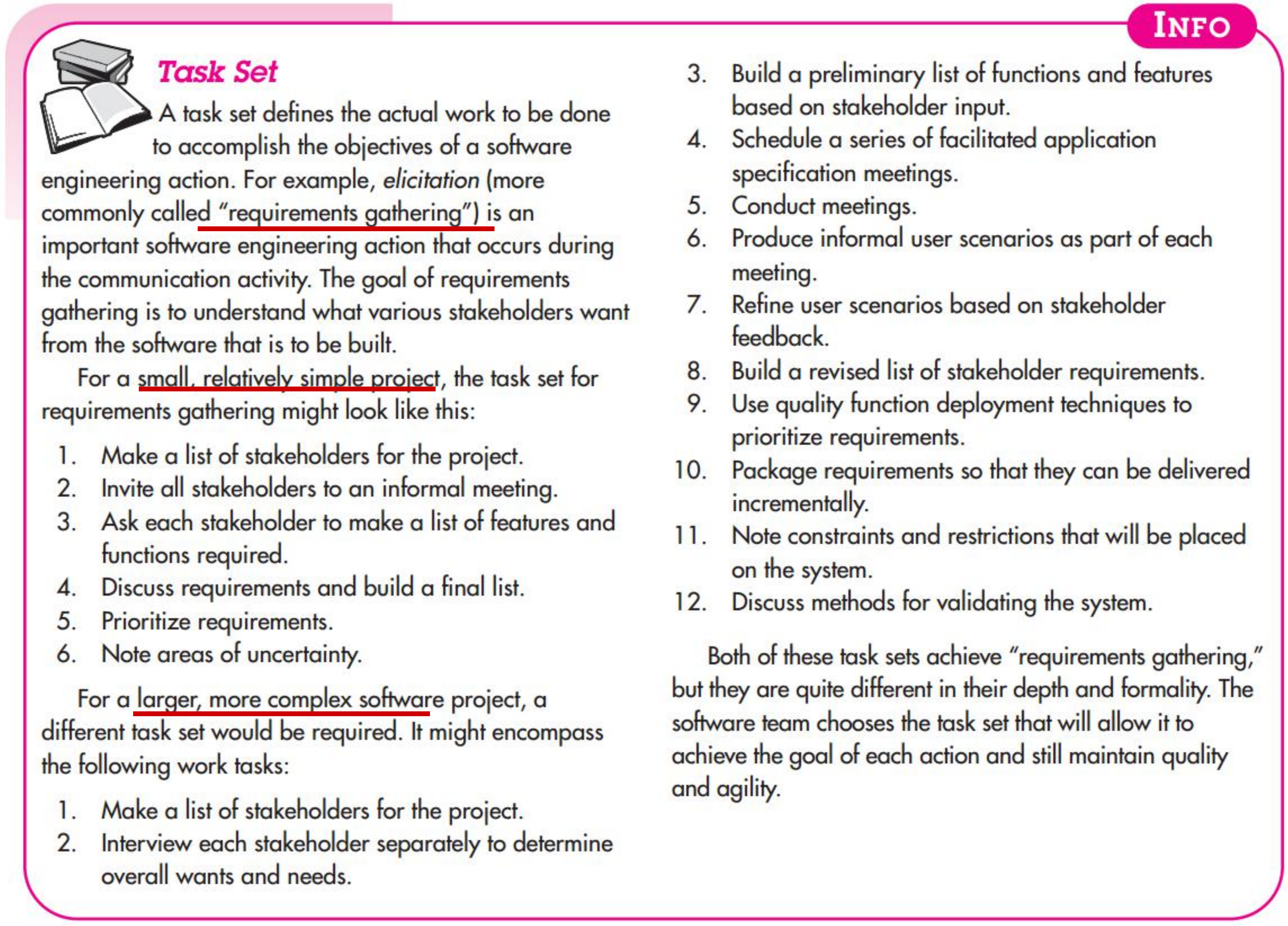
1. Make contact with stakeholder via telephone.
2. Discuss requirements and take notes.
3. Organize notes into a brief written statement of requirements.
4. E-mail to stakeholder for review and approval.

If the project was considerably more complex with many stakeholders, each with a different set .

3.3 Identifying a Task Set

- A task set defines the actual work to be done to accomplish the objectives of a software engineering action.
  - A list of the task to be accomplished
  - A list of the work products to be produced
  - A list of the quality assurance filters to be applied

3.3 Identifying a Task Set



INFO

Task Set

A task set defines the actual work to be done to accomplish the objectives of a software engineering action. For example, *elicitation* (more commonly called "requirements gathering") is an important software engineering action that occurs during the communication activity. The goal of requirements gathering is to understand what various stakeholders want from the software that is to be built.

For a small, relatively simple project, the task set for requirements gathering might look like this:

1. Make a list of stakeholders for the project.
2. Invite all stakeholders to an informal meeting.
3. Ask each stakeholder to make a list of features and functions required.
4. Discuss requirements and build a final list.
5. Prioritize requirements.
6. Note areas of uncertainty.

For a larger, more complex software project, a different task set would be required. It might encompass the following work tasks:

1. Make a list of stakeholders for the project.
2. Interview each stakeholder separately to determine overall wants and needs.
3. Build a preliminary list of functions and features based on stakeholder input.
4. Schedule a series of facilitated application specification meetings.
5. Conduct meetings.
6. Produce informal user scenarios as part of each meeting.
7. Refine user scenarios based on stakeholder feedback.
8. Build a revised list of stakeholder requirements.
9. Use quality function deployment techniques to prioritize requirements.
10. Package requirements so that they can be delivered incrementally.
11. Note constraints and restrictions that will be placed on the system.
12. Discuss methods for validating the system.

Both of these task sets achieve "requirements gathering," but they are quite different in their depth and formality. The software team chooses the task set that will allow it to achieve the goal of each action and still maintain quality and agility.

单选题 1分

设置

此题未设置答案，请点击右侧设置按钮

The cost of correcting an error made during the initial analysis of a project is estimated to be (　　) the cost of correcting a similar error after the system has been turned over to the customer.

A same

B 1/10

C 1/100

提交

3.4 Process Pattern

- How to define my own process framework?
- Is there efficient way to help development team build software products?

- It is process patterns.
- It is proven solutions to commonly encountered development problems.
- If developers can recognize that this is problem seen before they can use a previously known means of solving it, without taking the time to invent a new solution.

3.4 Process Patterns

- A *process pattern* **(summarized suggestion)**
  - describes a process-related problem that is encountered during software engineering work,
  - identifies the environment in which the problem has been encountered,
  - suggests one or more proven solutions to the problem.

- Stated in more general terms, a process pattern provides you with a *template* [Amb98]—a consistent method for describing problem solutions within the context of the software process.

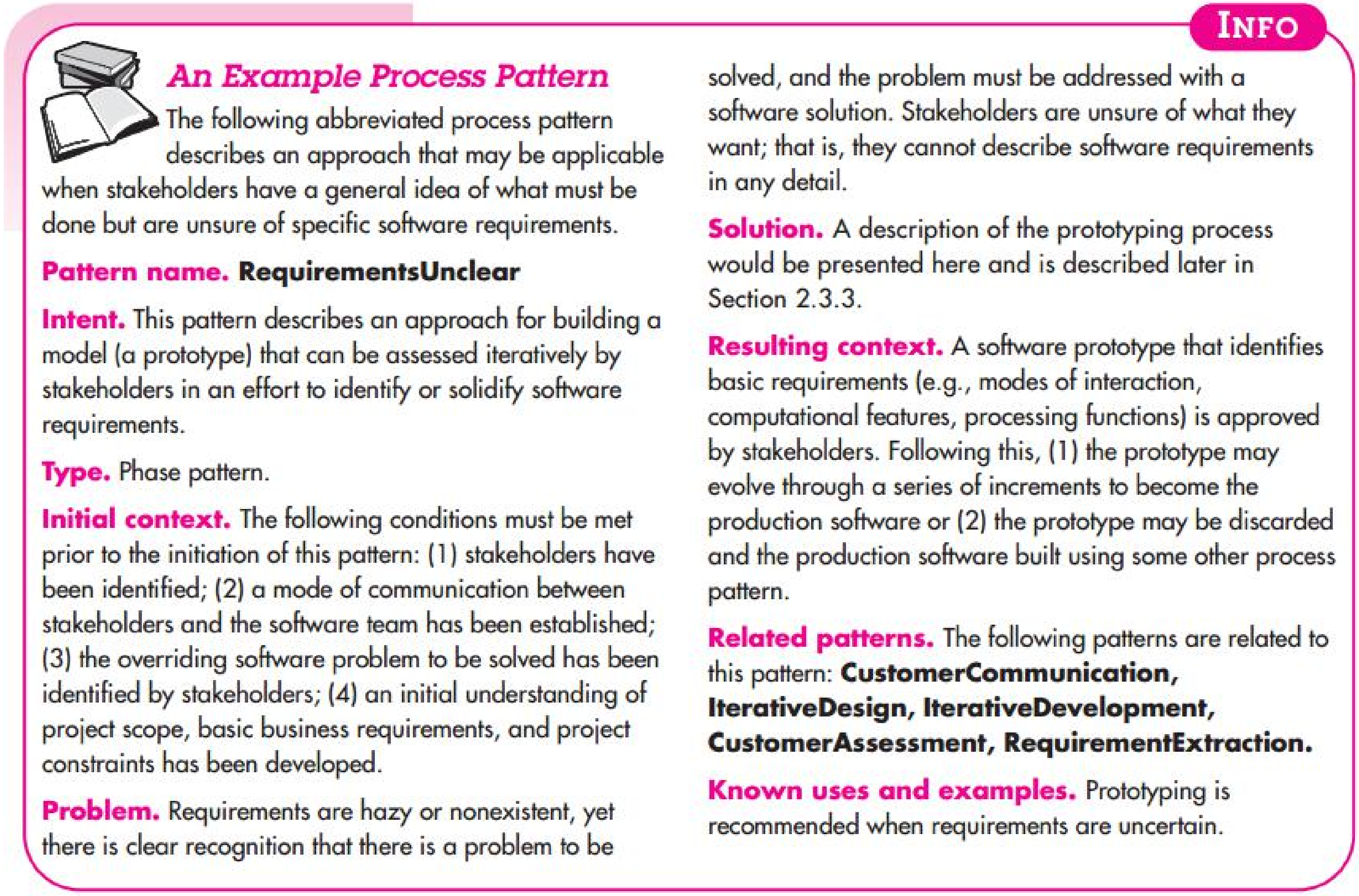
3.4 Process Pattern

- Pattern Name
- Forces
- Type
- Initial context
- Problem
- Solution
- Resulting Context
- Related Patterns
- Known Uses and Examples

3.4 Process Patterns Types

- *Stage patterns*—defines a problem associated with a framework activity for the process.
- *Task patterns*—defines a problem associated with a software engineering action or work task and relevant to successful software engineering practice
- *Phase patterns*—define the sequence of framework activities that occur with the process, even when the overall flow of activities is iterative in nature.

3.4 Process Pattern - Example



INFO

An Example Process Pattern

The following abbreviated process pattern describes an approach that may be applicable when stakeholders have a general idea of what must be done but are unsure of specific software requirements.

Pattern name. **RequirementsUnclear**

Intent. This pattern describes an approach for building a model (a prototype) that can be assessed iteratively by stakeholders in an effort to identify or solidify software requirements.

Type. Phase pattern.

Initial context. The following conditions must be met prior to the initiation of this pattern: (1) stakeholders have been identified; (2) a mode of communication between stakeholders and the software team has been established; (3) the overriding software problem to be solved has been identified by stakeholders; (4) an initial understanding of project scope, basic business requirements, and project constraints has been developed.

Problem. Requirements are hazy or nonexistent, yet there is clear recognition that there is a problem to be solved, and the problem must be addressed with a software solution. Stakeholders are unsure of what they want; that is, they cannot describe software requirements in any detail.

Solution. A description of the prototyping process would be presented here and is described later in Section 2.3.3.

Resulting context. A software prototype that identifies basic requirements (e.g., modes of interaction, computational features, processing functions) is approved by stakeholders. Following this, (1) the prototype may evolve through a series of increments to become the production software or (2) the prototype may be discarded and the production software built using some other process pattern.

Related patterns. The following patterns are related to this pattern: **CustomerCommunication, IterativeDesign, IterativeDevelopment, CustomerAssessment, RequirementExtraction.**

Known uses and examples. Prototyping is recommended when requirements are uncertain.

3.5 Process Assessment and Improvement

- **Standard CMMI Assessment Method for Process Improvement (SCAMPI)** — provides a five step process assessment model that incorporates five phases: initiating, diagnosing, establishing, acting and learning.
- **CMM-Based Appraisal for Internal Process Improvement (CBA IPI)**—provides a diagnostic technique for assessing the relative maturity of a software organization; uses the SEI CMM as the basis for the assessment [Dun01]
- **SPICE—The SPICE (ISO/IEC15504)** standard defines a set of requirements for software process assessment. The intent of the standard is to assist organizations in developing an objective evaluation of the efficacy of any defined software process. [ISO08]
- **ISO 9001:2000 for Software**—a generic standard that applies to any organization that wants to improve the overall quality of the products, systems, or services that it provides. Therefore, the standard is directly applicable to software organizations and companies. [Ant06]

西北工業大學
NORTHWESTERN POLYTECHNICAL UNIVERSITY

THE END