

Unit I: Software Development Process (Weightage -12 marks)

① Define :- i) Software ii) Software engineering
⇒ Software

- Software is set of instruction, data or programs used to operate computers and execute specific tasks and provide output.

- In simpler words, we can't touch or feel the software.

- It is opposite of hardware.

- Software is defined as collection of computer program, procedures, rules and data.

Software Engineering:

- The branch of Computer Science that deals with design, development, testing and maintenance of software application.

- It is process of developing, testing and deploying computer application to solve real world problems by adhering to set of set of engineering principles.

- It is systematic and disciplined approach.

② Characteristics of Software (2/4 m)

→ Software is a set of instruction, data or program used to operate computers and execute specific task and provide output.

i) Software is developed or engineered,

it is not manufactured in classic sense.

- making software and making hardware might be seem similar, but they are mostly different.

- Software is coding and programming while hardware is physical stuff like computer and phones.

- Both need good design. But in hardware problem can be arised during manufactured but opposite software issues can be fix easily.

- People are important for both software and hardware. Software about coding and hardware Physical things.

- In software most of expenses come from work of engineers like programmers unlike building a house.

ii) Software doesn't wear out.

- Software doesn't get "old" like physical things.

- It remains usable for as long as it meet your needs.

- Once you have made software, you

can keep using as long as it serves. For example, listen to favorite song again and again.

- Instead of wearing out, software get update to improve the performance or add new feature.

iii) Most of software is custom built rather than being assembled from existing components.

- Engineers use reusable components so they can focus on creating new and innovative parts of design, rather than redoing what already have been done.

- Reusing existing component is common, to make development faster and efficient.

- Example, GUI (Graphical User Interface)

- Software component should be designed and implemented in such a way that it can be reused in different program.

- Various versions of software are possible new modifications or updatons are done in old software to form new software.

③ Software Engineering as a layered technology.
or
(4m)

List or draw layers of Software Engg. (2m)
⇒ Software engineering is a layered technology. That means, to develop software one will have to go from one layer to another.

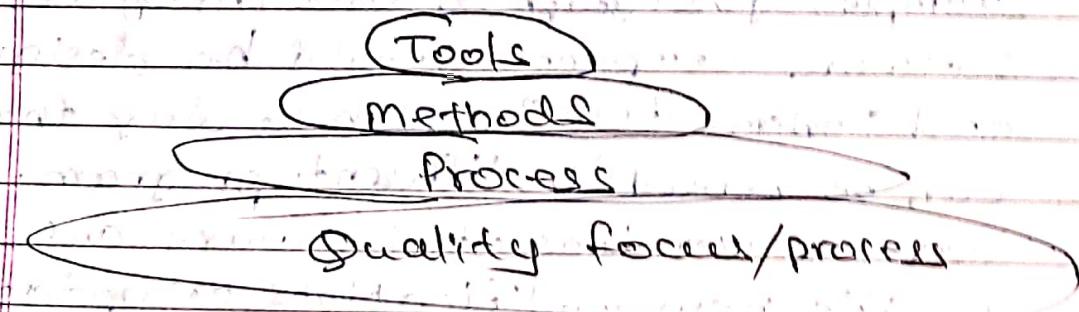
Any software can be developed using these layer approach.

The layers can related and each layer demands the fulfillment of previous layer.

There are 4 layers.

- Tools
- Methods
- Process
- Quality focus.

(TMPS)



Quality focus

When we build software, we want it to be of good quality, just how we like well made product.

- Quality is strong foundation of building.
- It holds everything together.
- Quality layer is Bedrock or Back Backbone.

- There are two types of Quality,
 - make sure that software works properly (functional)
 - it is easy to keep working well, (Non-functional).
- Just like we keep making things better in life, in SEN, we use methods like total quality management and six sigma to keep better at what we do.

Process Layer (what to do)

- It is a base layer or foundation layer for Software Engineering.
- Process layer is a key to keep all levels together.
- It includes framework that include different activities and task.
- In short, it covers all activities, actions and tasks required to carry out software development.

Methods (How to do)

- Software methods are like step-by-step guides for making software.
- These guides cover everything from talking to people to testing & support of software.
- It provides different techniques for building software and fixing issues.
- It helps deciding the things like which language to use and how different part will work together.

Tools

- Software tools making building software easier.
- They can work by themselves or with bit of help.
- When tools share information, it's called computer-aided software engineering.
- Some tools do work by themselves, other need bit of help. Together they like a team of computer working on software.
- Examples, editor, Database and Testing.

④ List of types of Software : (2 m)

or

Explain types of Software : (4 m).

→ Software is a set of instruction, data or programs used to operate computers and execute specific task and provide output.

Types of software

→ System Software

→ Application Software

→ A-I Software

→ Web-Based Software

→ Embedded Software

→ Real-Time Software

→ Engineering Software

✓ System Software

System software is program designed to run a computer's hardware and applications and manage its resources, such as its memory, processors and devices. It also provides a platform for running application software, and system software is typically bundled with a computer's operating system.

✓ Application software

Application software is a type of computer program that performs a specific personal, educational and business function.

Each application is designed to assist end-users in accomplishing variety of tasks, which may be related to productivity, creativity or communication.

✓ AI Software

Artificial Intelligence (AI) refers to computer systems capable of performing complex tasks that historically only a human could do, such as reasoning, making decisions or solving problems.

AI is the simulation of human intelligence processes by machines, especially computer systems.

Web-Based software

Web-Based software is software you access with just an internet connection and a web browser.

There is no software or hardware to purchase, no need to download software or ever worry about costly product upgrades.

Embedded software

Embedded software is used to control the limited set functions of hardware devices and doesn't generally need input; it is not typically worked with directly by users.

Its functions are activated by external controls, either external actions of the device itself or remote input.

Real-Time Software

A real-time application, or RTA, is an software / application that functions within a time frame that the user sees as immediate or current.

The use of real-time software is part of real-time computing.

Common examples: Air traffic control systems

process control systems

Autonomous driving system.

Engineering/Scientific Software

It is software satisfy the needs of scientific or engineering user to perform enterprise specific task.

Such software is written for specific applications using principles, techniques and formulae specific to that field.

Examples, AUTOCAD, PSPICE, ORCAD, etc.

(5) Explain process framework? (4m/6m)

⇒ Framework is a standard way to build and deploy applications.

Software Project framework is a foundation of complete Software engineering process.

Software process framework includes all set of umbrella activities.

It also includes number of framework activity that are applicable to all Software projects.

Process framework

Umbrella activities

Frameworks activities

Task Sets

Task

Work

ISQA Points

A generic process framework identifies encompasses five activities which are given below one by one.

① Communication

In this activity, heavy communication with customers and other stakeholders requirement gathering is done.

② Planning

In this activity, we discuss the technical related tasks, work schedule, risks, required resources etc.

③ Modelling

Modelling is about building representations of things in 'real-world'.

In modelling activity, a product's model is created in order to better understand the requirements.

④ Construction

In software engineering, we generate the code and test the product in order to make better product.

⑤ Deployment

In this activity, a complete or non-complete products or software are presented to the customers to evaluate and give feedback on the basis of their feedback. On the basis of their feedback we modify the products for supply better product.

Umbrella activities

Software engineering is a collection of interconnected phases. These steps are expressed or available in different ways in different software process models.

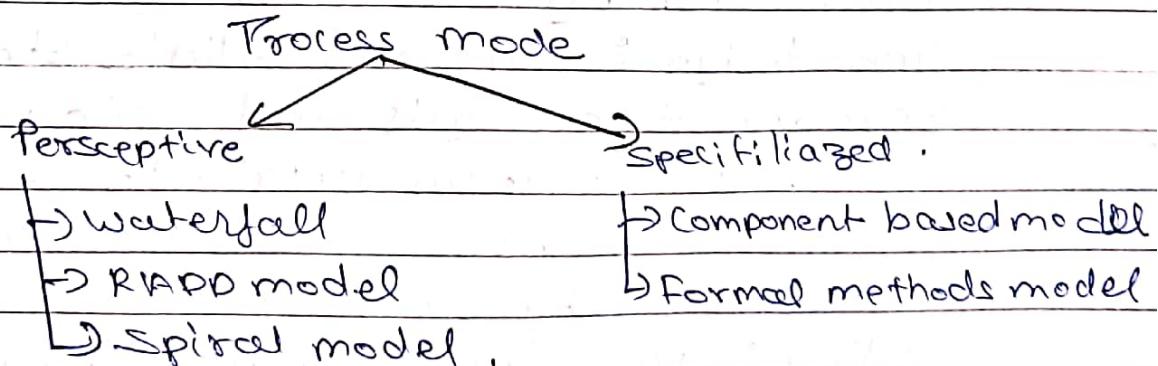
Umbrella activities are a series of steps or procedures followed by a software development team to maintain the progress, quality changes and risk of complete development tasks.

Umbrella activities will evolve through the phases of generic view of software.

⑥ Describe ~~waterfall model~~ ^{Process} model? (2m)

→ Process model

- The process model can be defined as the abstract representation of process.
- The appropriate process model can be chosen based on abstract representation of process.
- The software process model is also known as Software Development Life Cycle (SDLC) model or software paradigm.

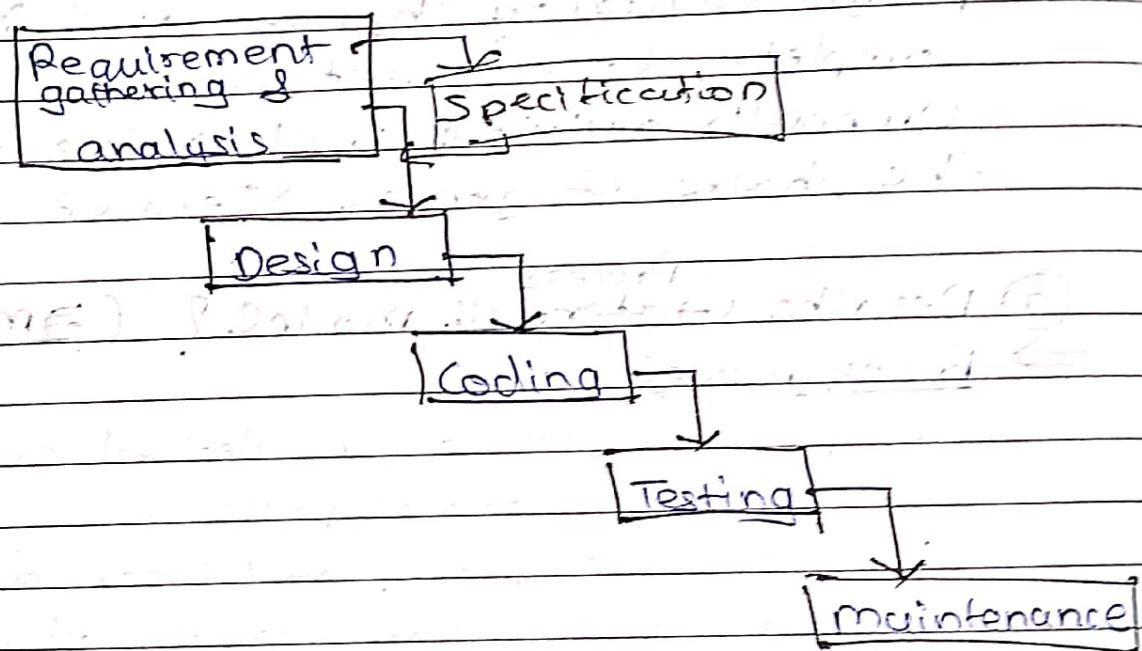


⑦ Describe and Explain waterfall model. (4m).

→ The waterfall model is also called as 'linear-sequential model' or 'classical life cycle model'.

The software development starts with requirements gathering phase. The process goes through analysis, design, coding, testing and maintenance.

Following figure shows Diagram of Waterfall Model.



- In requirement gathering and analysis phase the basic requirement of the system must be understood by software engineer.
- The design is an intermediate step between requirement analysis and coding. Design focus on program attribute such as Data structure, software architecture, Algorithm details.
- Coding is the step in which design is translated into code.

QAP: IBM 1980

90-60

Business model

Page No.:
Date: / /

↓
Design
modeling

Process modeling

↓ Model for the
turnaround

into machine-readable form. Programs are created in these phases.

- Testing begins when coding is done. The purpose of testing is to uncover errors, fix the bugs and meet customer requirements.
- Maintenance is longest life cycle phase. When we made software and handover to customer after that if any errors, updation and modification, additional of new features are done in this phase.

Advantages

Simple to implement

Only implement for smaller systems.

DisAdvantages

- There is no consideration for error correction.
- Delay the testing until the end of development is common.
- High amount of risk.

* Incremental model

Requirement analysis

Design

Implementation

Testing

Deployment

Maintenance

⑧ Distinguish between waterfall model and spiral model. (4m)

Page No. 1
Date 11



Waterfall Model

The waterfall mode is simple & easy.

The waterfall model works in sequential model method

Errors and risks are identified and rectified after completion of stage.

The waterfall mode is adopted by customers.

Applicable for small projects.

Flexibility to change in waterfall model is difficult.

High amount of risk

Compared less expensive

Customer involvement is less / minimum.

Spiral Model

The spiral model is lot more complex

Spiral mode works in evolutionary method.

Errors and risks are identified earlier.

Spiral mode is adopted by developers.

Applicable for larger projects.

Flexibility to change in spiral mode is not difficult

low amount of risk

costly

Customer involvement is maximum.

It require least maintenance.

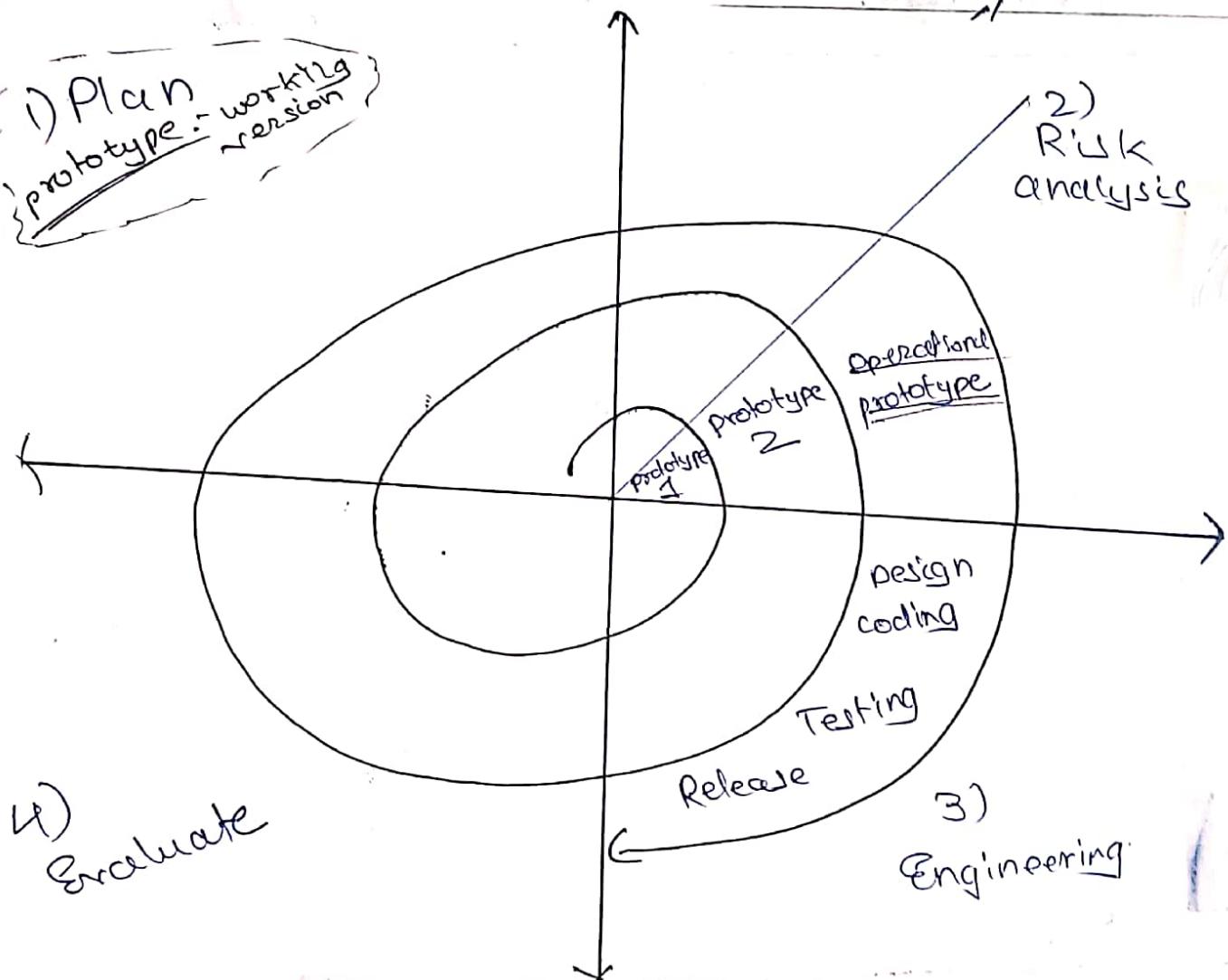
It require typical maintenance.

Based on linear framework type

Based on linear or iterative framework type

Reusability is extremely unlike reusability is possible.

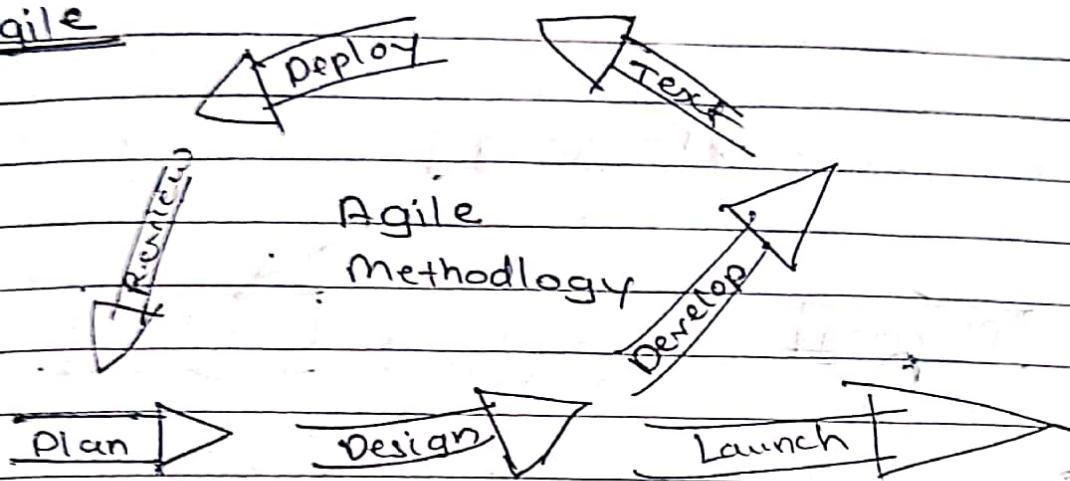
Spiral model



Q Distinguish between prescriptive process model and agile process model. (4M).

Prescriptive Model	Agile process model
Prescriptive process models stress detailed definitions, identifications, and applications of process activities and tasks.	Agile process model emphasize project "agility" and follow a set of principles that lead to more informal approaches to software process.
It is process oriented. It follows Life cycle model (waterfall, spiral) development model.	It is people oriented. It follows Iterative and Incremental development model.
Predictive planning is required Customer role is important.	Adaptive planning is required. Customer role is crucial.
Formal communication is required	Informal communication is required.
Documentation required is to be comprehensive and constant.	Documentation required is to be minimal and evolving.
Inexpensive	Expensive.
Examples: waterfall, spiral	Examples: Extreme programming, Scrum.

Agile



(10) Describe Extreme programming with proper diagram. (4m). ***

→ Extreme programming is a lightweight, efficient, low-risk, flexible, predictable, scientific and fun way to develop software.

Extreme programming (XP) was conceived and developed to address the specific needs of software development by small teams in face of vague and changing requirements.

Extreme Programming is one of Agile software development methodology.

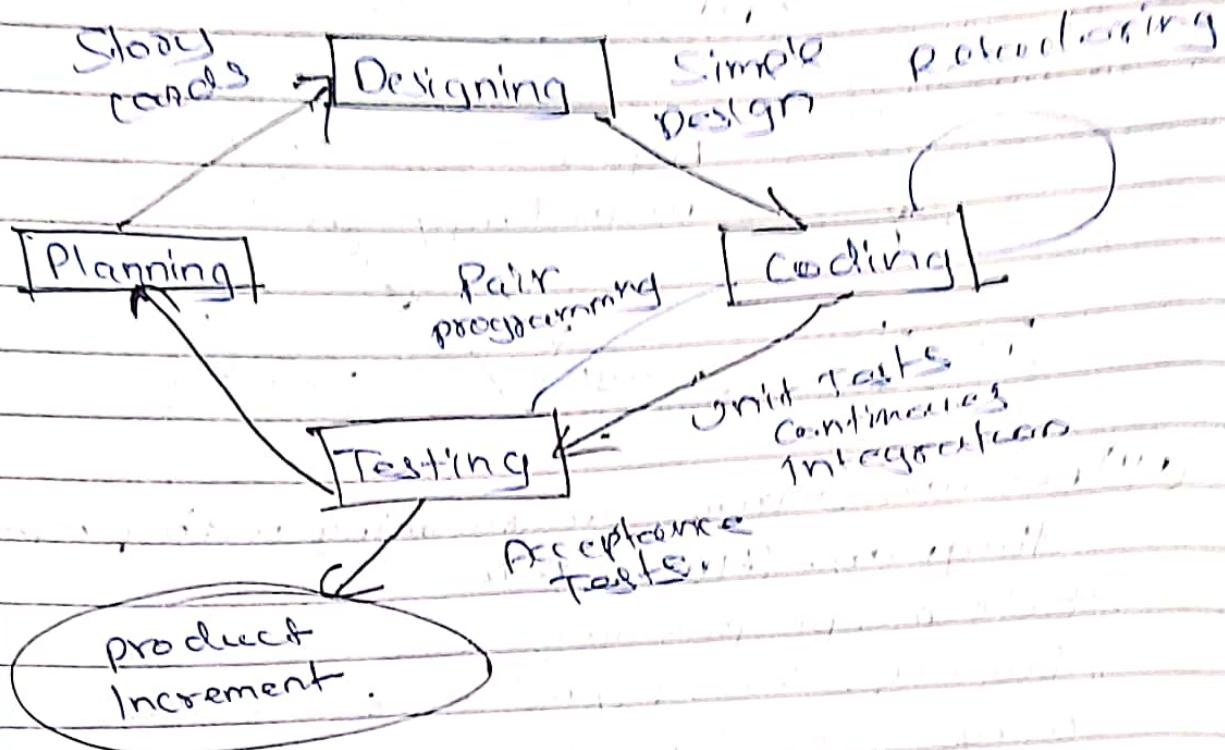
It provides values and principles to guide the team behaviour.

The team is expected to self organize.

Extreme programming provides specific core practices.

Where,

- ↳ Each practice is simple and self-complete
- ↳ Combination of practice produces more complex and emergent behaviour.



Extreme programming is based on following values.

- Communication
- Simplicity
- Feedback
- Courage
- Respect

Explain - - - - -

Extreme Programming solves following problem often faced in software development projects.

- Testing is effective as there is continuous regression testing.
- Design is effective as everybody needs to do refactoring.
- Integration testing is important and test several times.
- Slipped schedules: Short and achievable development cycles ensure timely delivery.
- Business changes: Changes are inevitable and are accommodated at any point of time.

- Cost incurred in changes: Extensive and ongoing testing make sure the changes do not break the existing functionality.
- Production and post-delivery defects: Emphasis on unit tests to detect and fix the defects early.
- Misunderstanding the business and/or domain: Making the customer a part of the team ensures constant communication and clarifications.

(11) List any four selection criteria for Software Process model. (2m).

→ Waterfall: If a small project is to be implemented.
RAD: when project requirements are not clear.
Spiral: The risk of long project is not affordable.
Agile: when project requirements are not properly known.

(12) Draw failure curve of software

