Parul® University

Introduction to Software Engineering (203124253)

Prof. Nileshkumar Kakade, Assistant Professor
Computer Science & Engineering Engineering

Parul® University

TEXT BOOKS

Software Engineering: A Practitioner's Approach, by R.S.Pressman published by TMH.

Reference Books:

- **Software Engineering, 8th Edition by Sommerville, Pearson.**
- **Software Engineering 3rd Edition by Rajiv Mall, PHI.**
- **An Integrated Approach to Software Engineering by Pankaj Jalote Wiley India, 2009.**

- **Always Check notes**

Parul®
University

DIGITAL LEARNING CONTENT

CHAPTER-1

Introduction

ENGINEERING

- Definition of Engineering

-Application of science, tools and methods to find cost effective solution to problems

- Definition of SOFTWARE ENGINEERING

- SE is defined as systematic, disciplined and quantifiable approach for the development, operation and maintenance of software

Image source : Google

THE CHANGING NATURE OF SOFTWARE

- Seven Broad Categories of software are challenges for software engineers
- System software
- Application software
- Engineering and scientific software
- Embedded software
- Product-line software
- Web-applications
- Artificial intelligence software

Image source : Google

LEGACY SOFTWARE

- Older programs
- Developed decades ago.
- The quality is poor
- Inextensible design
- Convoluted code
- Poor and nonexistent documentation
- Test cases and results are not achieved.
- E.g. 25+ years old software in planes is still in use.

Quiz

What type of software is used in this ?

Image source : https://pixabay.com/

Software Characteristics

1. Software is developed or engineered
2. not manufactured
3. Software doesn't "wear out."
4. Although the industry is moving toward component-based assembly, most software continues to be custom built.

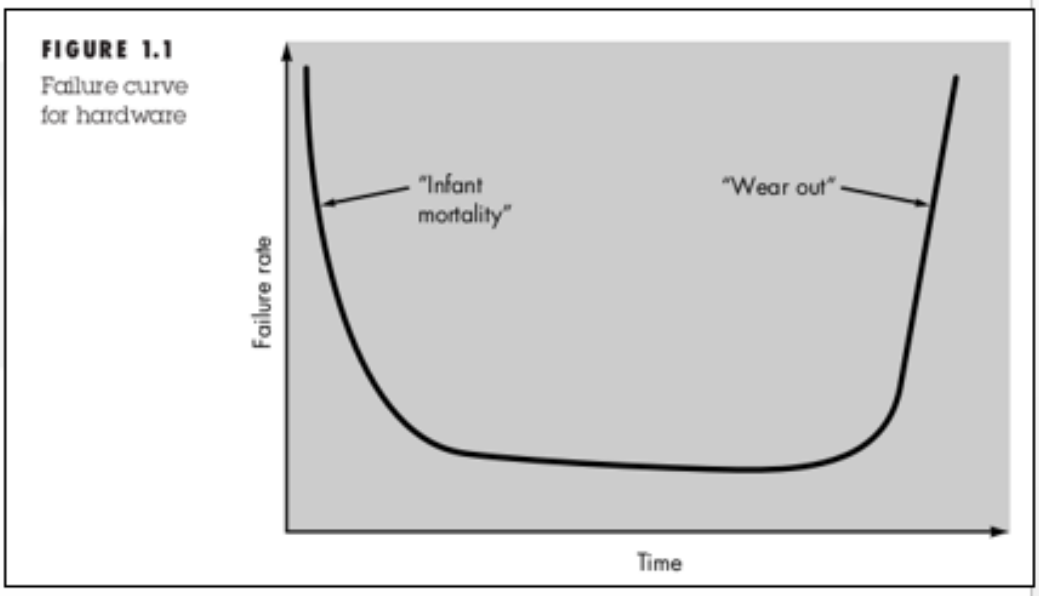


FIGURE 1.1
Failure curve for hardware

Image source : Pressman, R. S. (1987). *Software engineering: A practitioner's approach*. New York: McGraw-Hill.

Software Characteristics

FIGURE 1.2
Idealized and actual failure curves for software

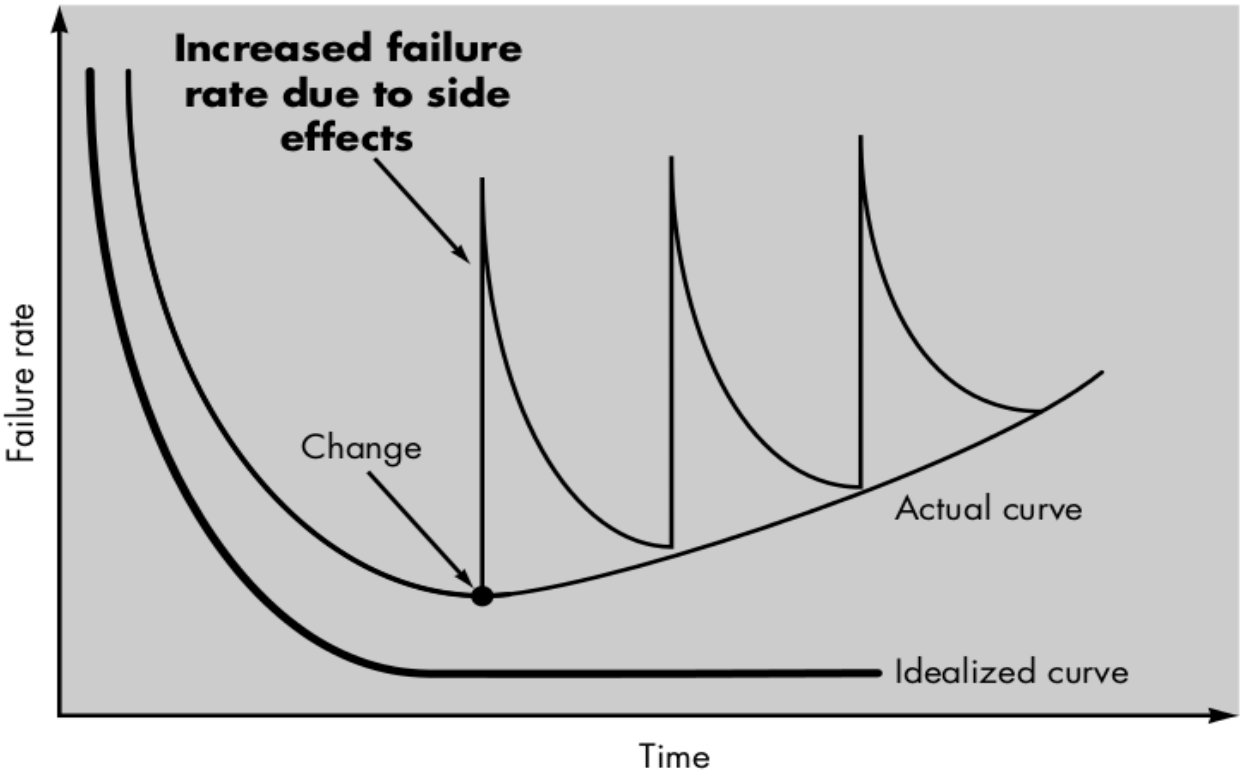


Image source : Pressman, R. S. (1987). *Software engineering: A practitioner's approach*. New York: McGraw-Hill.

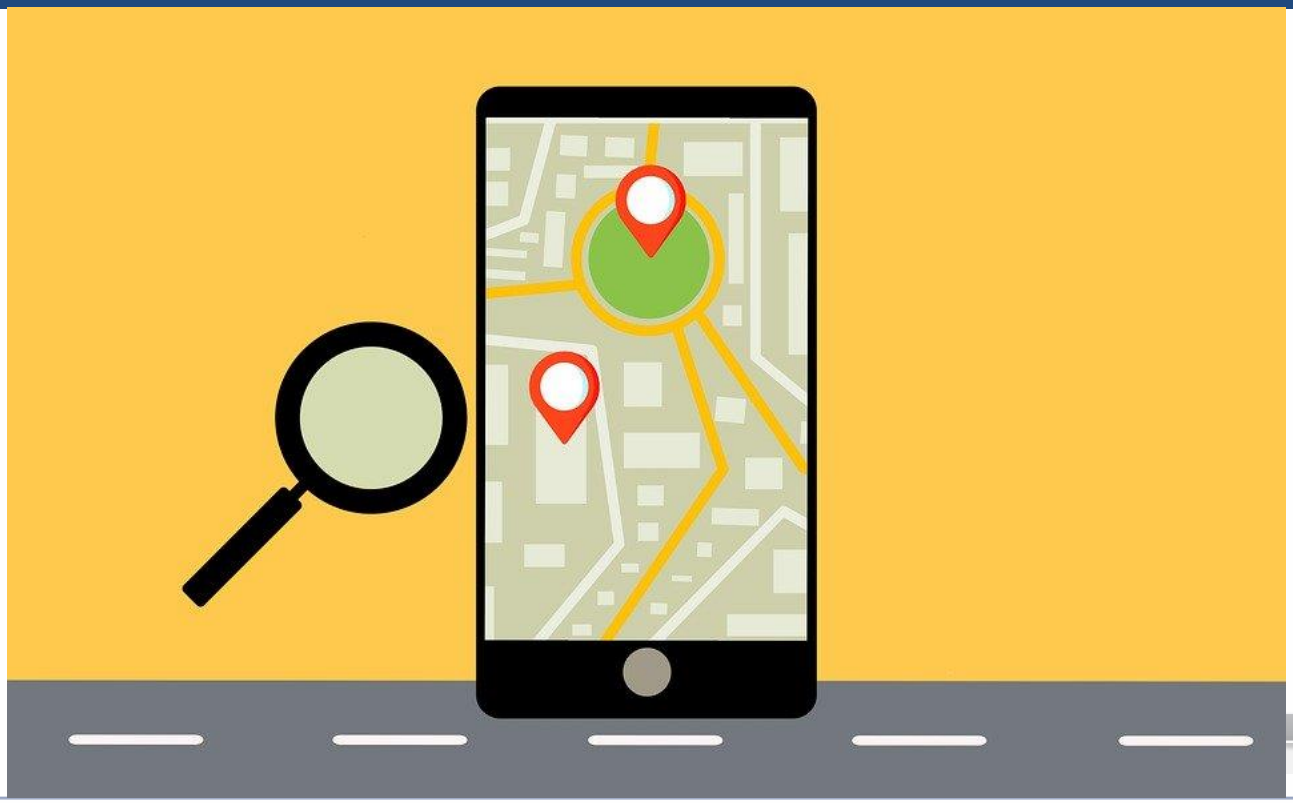
KEY POINT
Software engineering methods strive to

Software Applications

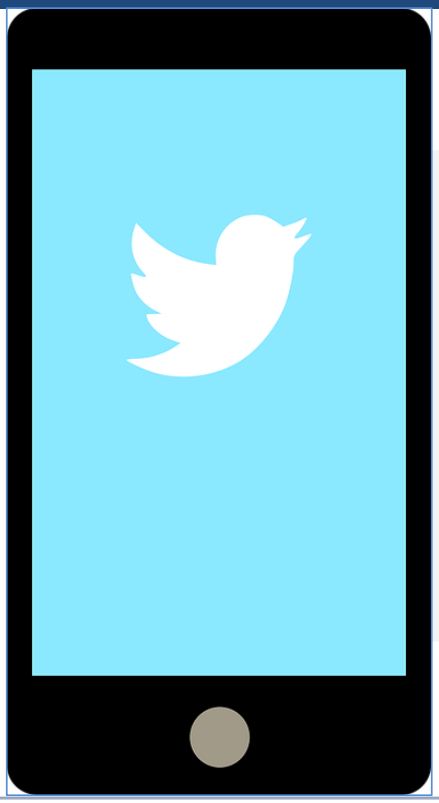
1) System software
2) Real-time software
3) Business software
4) Engineering and scientific software
5) Embedded software
6) Personal computer software
7) Web-based software
8) Artificial intelligence software

Quiz

Type of application?



Type of application?



Type of application?

Image source : PUMIS>IN

A Layered Technologies

Generic View Of Software Engineering

- The work associated with software engineering can be categorized into three generic phases, regardless of application area, project size, or complexity.
- The definition phase: focuses on what, identify what information is to be processed, what function and performance are desired, what system behavior can be expected, what interfaces are to be established, what design constraints exist, and what validation criteria are required to define a successful system.
- The key requirements of the system and the softwar major tasks
- 1. system or information engineering
- 2. software project planning
- 3. requirements analysis

Think about MIS

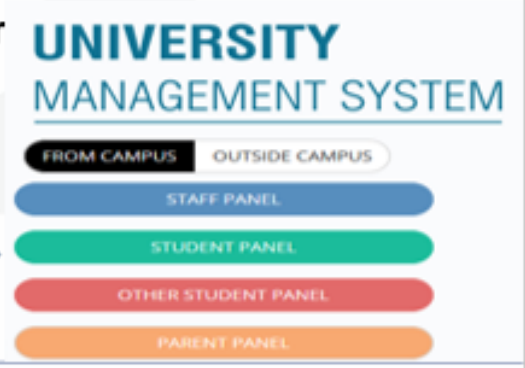


Image source : Google

Cntd....

- The **development phase** focuses on how.
- define how data are to be structured
- how function is to be implemented within a software architecture
- how procedural details are to be implemented
- how interfaces are to be characterized
- how the design will be translated into a programming language (or nonprocedural language)
- how testing will be performed.
- three specific technical tasks : 1. software design (Chapters 13–16, and 22) 2. code generation 3. software testing (Chapters 17, 18, and 23).

Image source : Google

Cntd....

The **support phase** focuses on change associated with error correction, adaptations required as the software's environment evolves, and changes due to enhancements brought about by changing customer requirements.

Four types of change are encountered during the support phase :

Correction: customer will uncover defects. Corrective maintenance changes will be done

Adaptation: results in modification to the software to accommodate changes to its external environment.

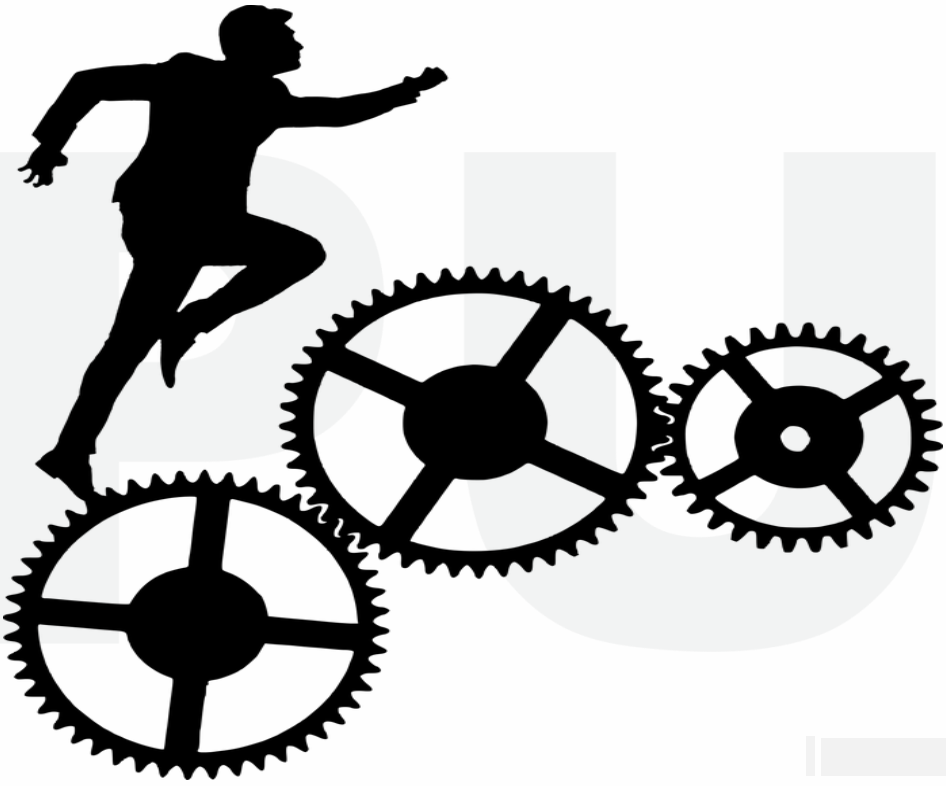
Enhancement: Perfective maintenance extends the software beyond its original functional requirements.

Prevention: enable the software to serve the needs of its end users.

In essence, preventive maintenance makes changes to computer programs so that they can be more easily corrected, adapted, and enhanced.

Image source : Google

Process Models

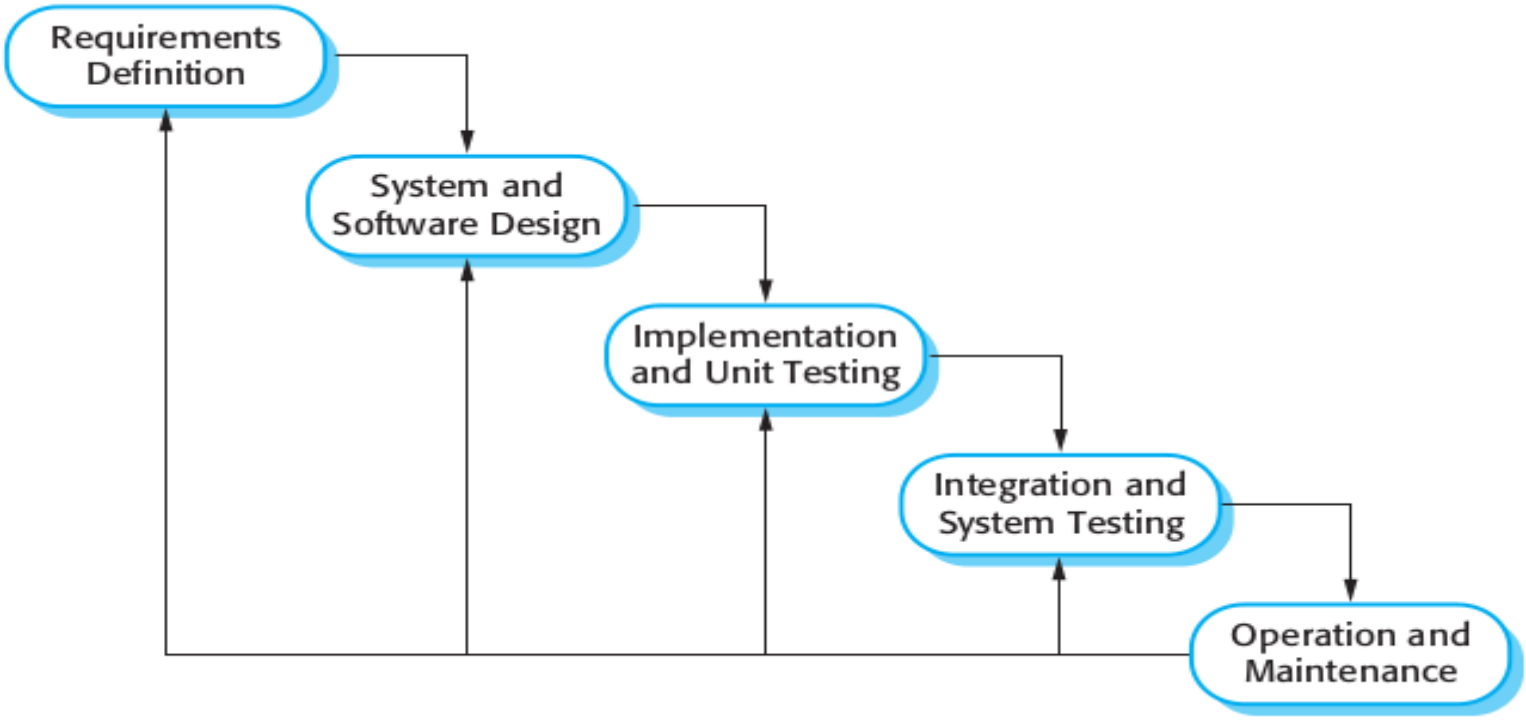


Process Model

- To solve actual problems in an industry setting, a software engineer or a team of engineers must incorporate a development strategy that encompasses the process, methods, and tools layers
- A process model for software engineering is chosen based on :
  - The nature of the project and application
  - The methods and tools to be used
  - The controls and deliverables that are required

Image source : Google

Waterfall Model



•Image source : *Ian Sommerville. 2010. Software Engineering (9th. ed.). Addison-Wesley Publishing Company, USA.*

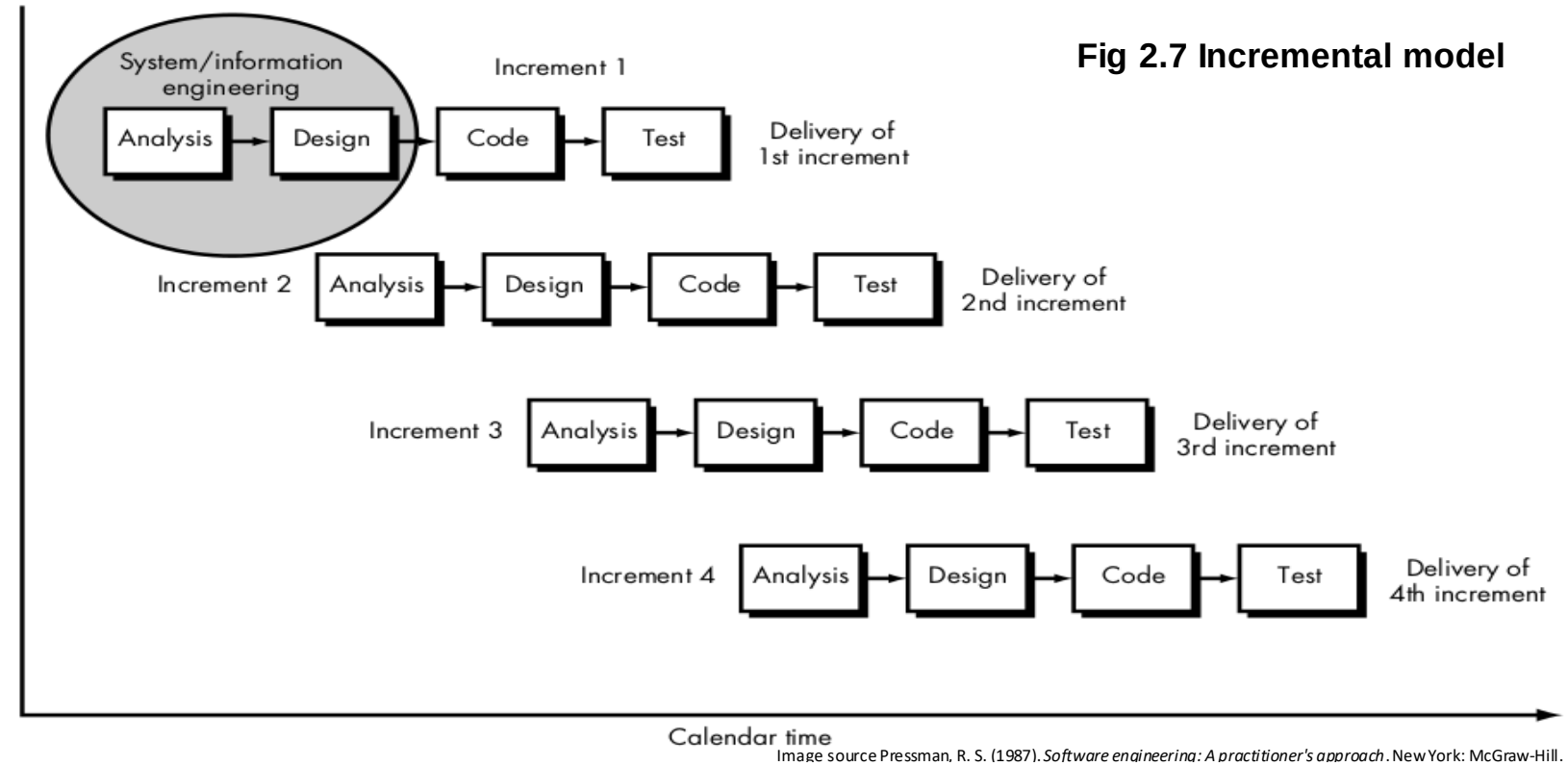
Waterfall model problems

- Inflexible partitioning
- difficult to respond to changing customer requirements
- Only appropriate when changes are least expected & requirements are well-understood
- Mostly used for large systems ,where a system is developed at several sites.
- In those circumstances, the plan-driven nature of the waterfall model helps coordinate the work.

Evolutionary Software Process Model

- Evolutionary models are iterative. They are characterized in a manner that enables software engineers to develop increasingly more complete versions of the software.
  - Incremental Model
  - Spiral Model
  - Concurrent Development Model

Image source : https://pixabay.com/illustrations/gps-map-phone-direction-smartphone-5137225/



Incremental

Fig 2.7 Incremental model

Image source Pressman, R. S. (1987). *Software engineering: A practitioner's approach*. New York: McGraw-Hill.

Understanding prototype

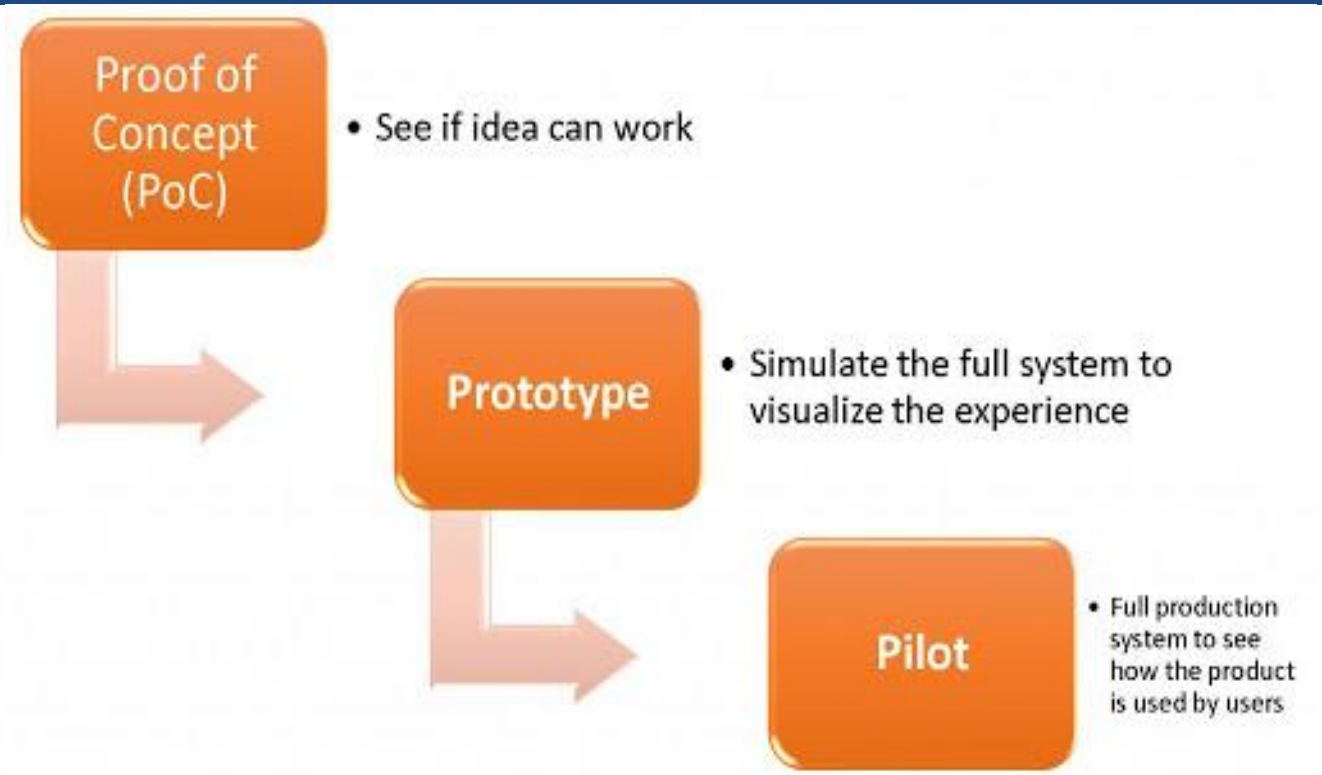


Image source Pressman, R. S. (1987). *Software engineering: A practitioner's approach*. New York: McGraw-Hill.

Prototyping

- Initial version
- Used to demonstrate concepts
- Try out design options
- Find out more about the problem and solutions.
- Rapid, iterative development
- costs are controlled
- system stakeholders can experiment

Prototyping

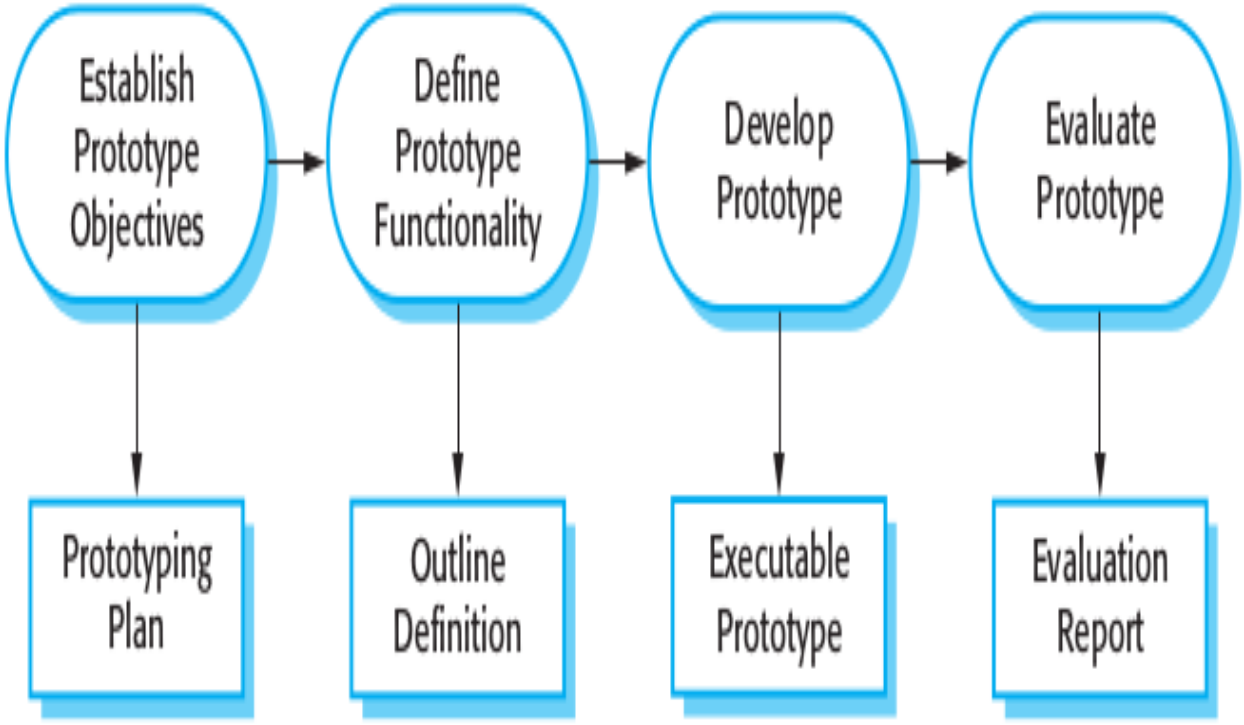


Figure 2.9 The process of prototype development

•Image source : *Ian Sommerville. 2010. Software Engineering (9th. ed.). Addison-Wesley Publishing Company, USA.*

Prototyping

- Requirements gathering
- Quick Design
- Prototype is constructed
- Prototype is evaluated by the customer/user
- The prototype can serve as "the first system."

Limitations

- Software quality or long-term maintainability is not considered
- Developer often makes implementation compromises
- Customer and developer must both agree on it
- It is discarded after use

Think ! Think ! Think !

If drones are designed to deliver material to hill areas/COVID19 patients , then what features should be included in software of prototype?

•Image source :https://pixabay.com/vectors/drone-photography-spying-2028611/

Spiral Model

- Software is developed in a series of incremental releases
- During early iterations incremental release might be a prototype. During later iterations, increasingly more complete versions of the engineered system are produced.
- A spiral model is divided into a number of framework activities, also called
- task regions –
  - Customer communication
  - Planning
  - Risk analysis
  - Engineering
  - Construction and release
  - Customer evaluation

Spiral Model

A typical spiral model

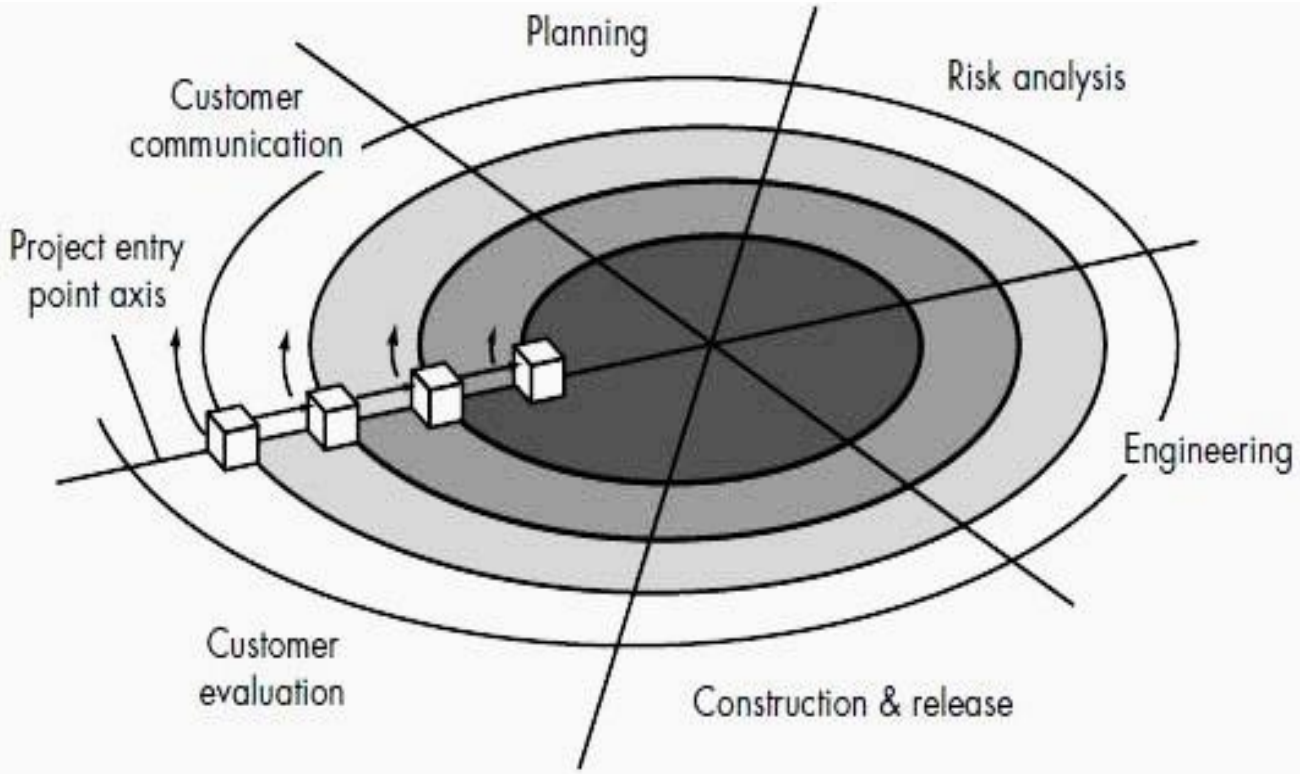
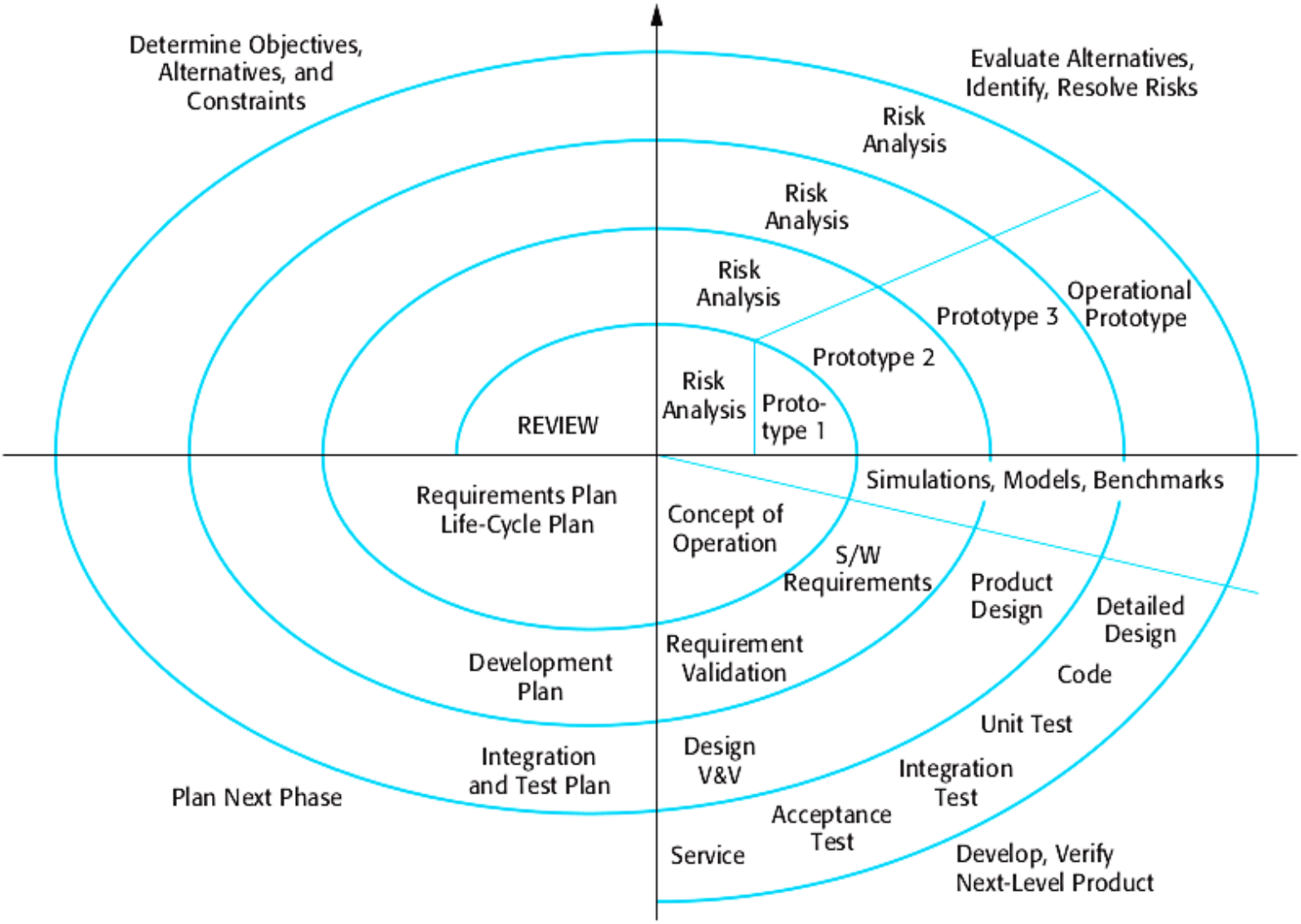


Image source Pressman, R. S. (1987). *Software engineering: A practitioner's approach*. New York: McGraw-Hill.



•Image source : *Ian Sommerville. 2010. Software Engineering (9th. ed.). Addison-Wesley Publishing Company, USA.*

Spiral Model

- **Benefits**
  - The developer and customer better understand and react to risks at each evolutionary level
  - It maintains the systematic stepwise approach suggested by the classic life cycle but incorporates it into an iterative framework that more realistically reflects the real world.
- **Limitation**
  - It may be difficult to convince customers (particularly in contract situations) that the evolutionary approach is controllable

concurrent process

- Series of major technical activities
- Series of tasks & their associated states.
- All activities exist concurrently
- All activities reside in different states.
- Events that will trigger transitions from state to state for activities.

Concurrent Development Model

FIGURE 2.10
One element of the concurrent process model

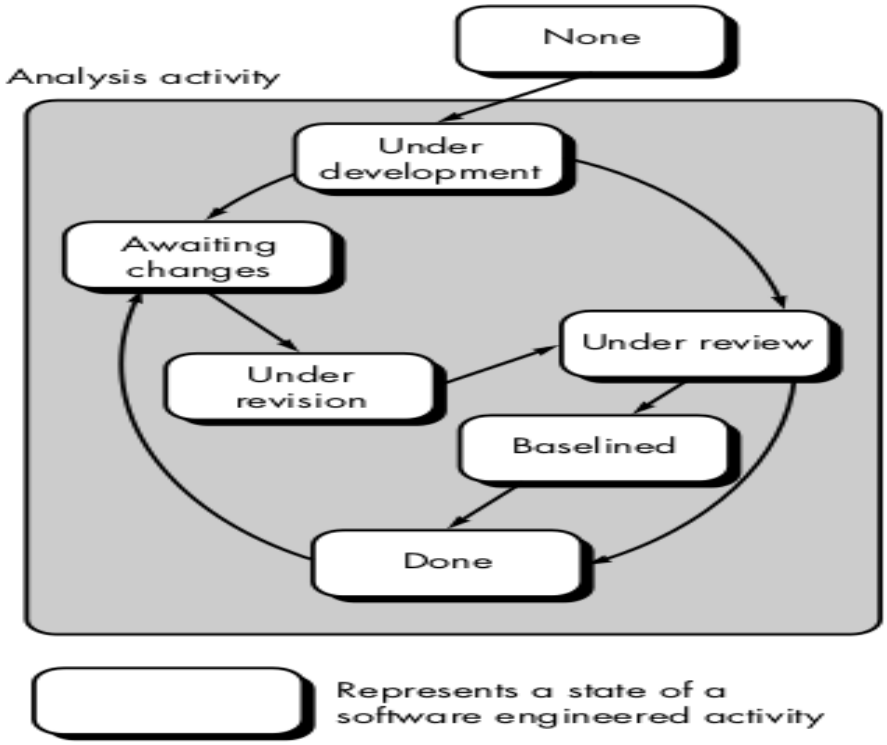
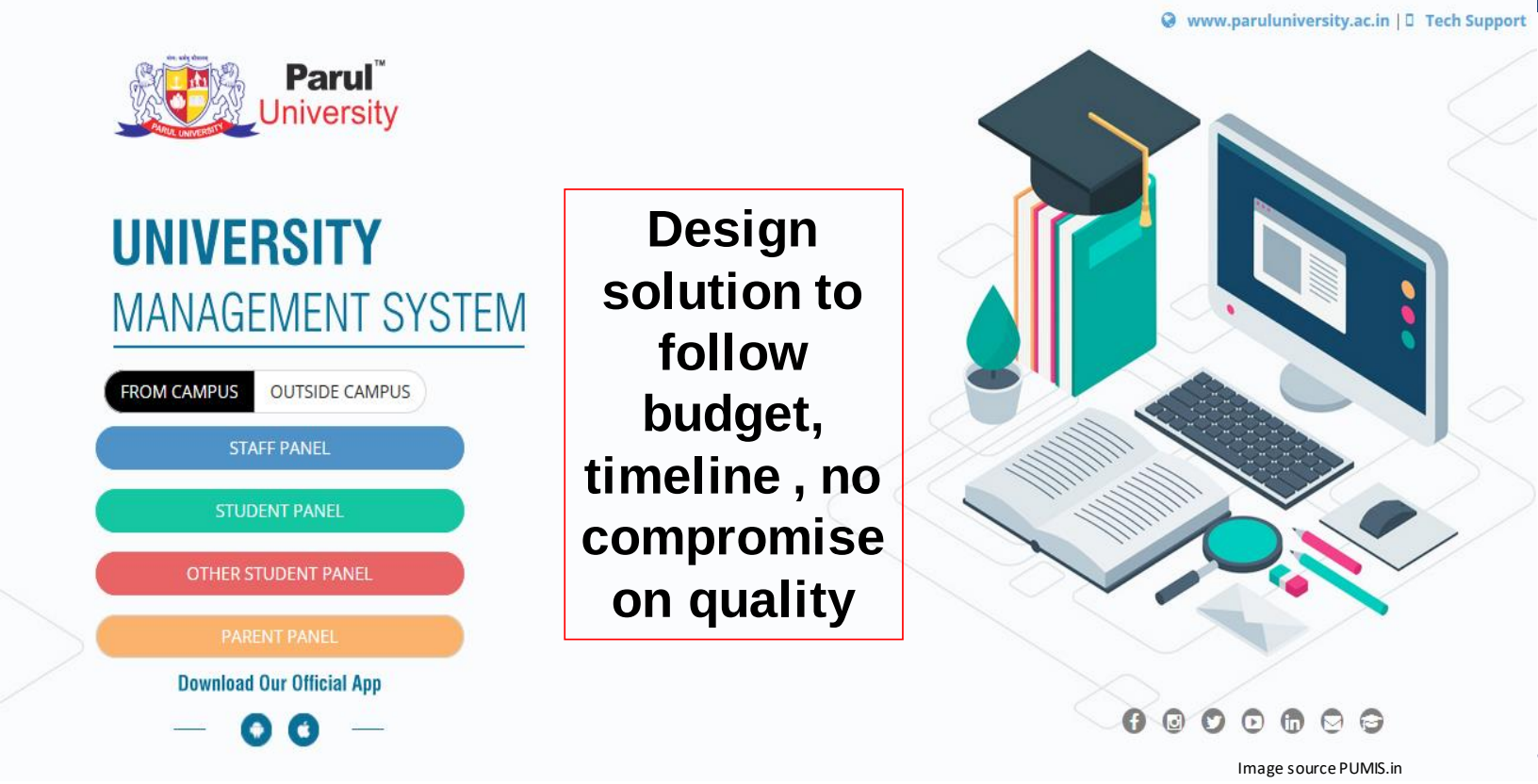


Image source Pressman, R. S. (1987). *Software engineering: A practitioner's approach*. New York: McGraw-Hill.

Which model should be used for development of PUMIS? Why?



Design solution to follow budget, timeline , no compromise on quality

Image source PUMIS.in

Rapid Application Development (RAD) Model

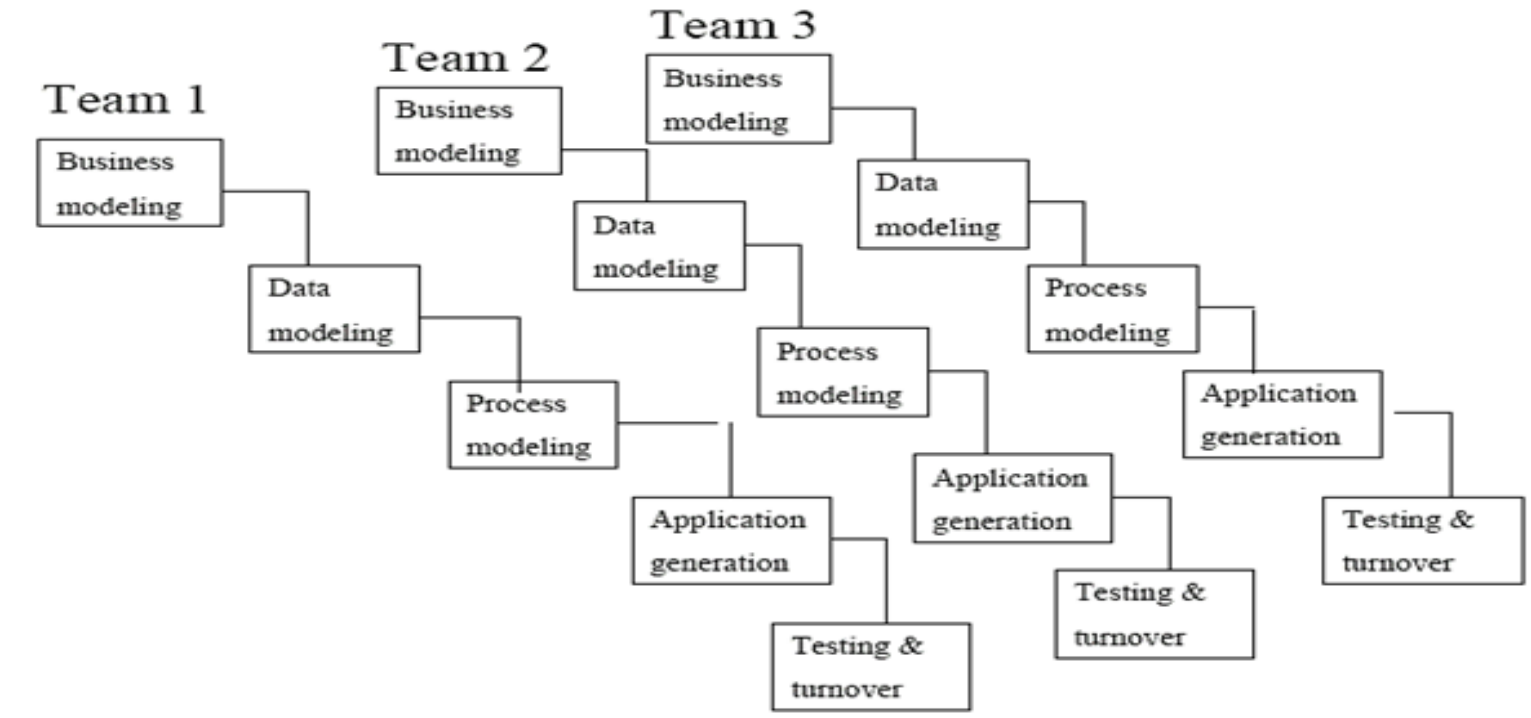


Image source : Google

Rapid Application Development (RAD) Model

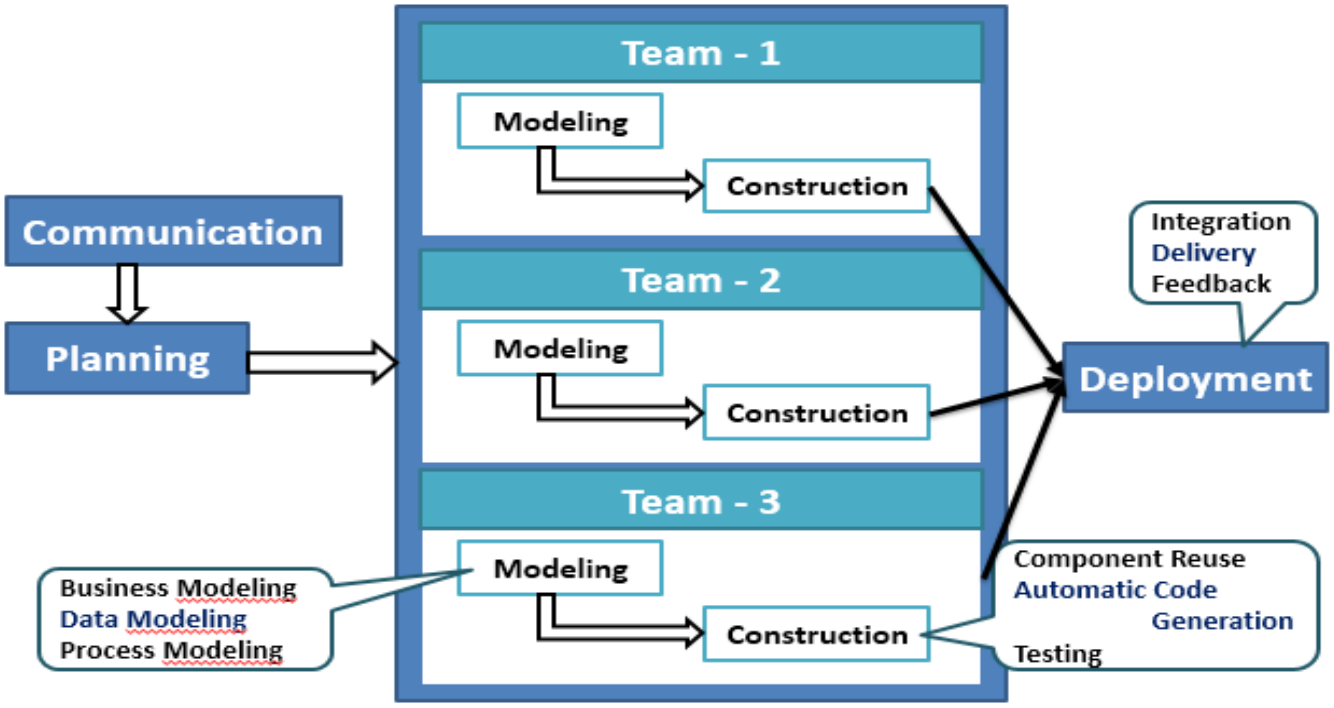


Image source : Google

Rapid Application Development (RAD) Model

- It is also know as **RAD** Model
- It is a type of **incremental model** in which; **components** or functions are **developed in parallel**.
- Rapid development is **achieved** by **component based construction**
- This can **quickly give** the customer **something to see** and use and to provide feedback.
- **Communication**
  - This phase is used to understand business problem.
- **Planning**
  - Multiple software teams work in parallel on different systems/modules.

Rapid Application Development (RAD) Model

- **Modeling**
  - **Business Modeling:** *Information flow* among the business.
    - Ex. What kind of information drives (moves)?
    - Who is going to generate information?
    - From where information comes and goes?
  - **Data Modeling:** Information refine into set of *data objects* that are *needed* to support business.
  - **Process Modeling:** *Data object* transforms **to** *information flow* necessary to implement business.
- **Construction**
  - It highlighting the *use of pre-existing software component*.
- **Deployment**
  - Deliver to customer basis on subsequent iteration.

Parul® University

DIGITAL LEARNING CONTENT

Agile Development

Image source: https://pixabay.com/vectors/yoga-exercise-gymnastics-stretch-304634/

Agility

- Agility is ability to move quickly and easily.
- Property consisting of quickness, lightness & ease of movement
- Ability to create and respond to **change**
- Ability to quickly reprioritize use of resources on requirements, technology, and knowledge shift
- A very fast response to sudden market **changes**
- Respond to threats by intensive customer interaction
- Use of evolutionary, incremental, and iterative delivery to converge on an optimal customer solution
- Maximizing BUSINESS VALUE with right sized, just- enough, and just-in-time processes and documentation

Agile Methodology

- Best suited where requirements usually **change** rapidly during the development process
- **#IMP: The Manifesto for Agile Software Development :**
- We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:
- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to **change** over following a plan

Change ! Change ! Change !

•Why there are changes in requirements? It is not necessary that it's a mistake. there are four fundamental sources of change [pressman]:

1. New business or market conditions dictate changes in product requirements or business rules.

Image source: https://pixabay.com/illustrations/icon-set-social-media-world-digital-1232558/

Change ! Change ! Change !

•2. New customer needs demand modification of data produced by information systems, functionality delivered by products, or services delivered by a computer based system.

•Instagram Recently added features:

- **Support Small Business**
- **Shops on Instagram**
- **Share Live Videos**
- **Better picture resolution**
- **New Download Your Data Feature**
- **New Select Partnerships Through Stories etc.**

Image source: https://pixabay.com/illustrations/instagram-insta-logo-new-images-1675670/

Change ! Change ! Change !

3. **Reorganization or business growth / downsizing causes changes in project priorities or software engineering team structure.**

4. Budgetary or scheduling constraints cause a redefinition of the system or product: money & time are most important factors to be taken care in any project.

Image sourcehttps://pixabay.com/vectors/bag-money-wealth-revenue-finance-147782/

Principle	Description
Customer involvement	Customers should be closely involved throughout the development process. Their role is provide and prioritize new system requirements and to evaluate the iterations of the system.
Incremental delivery	The software is developed in increments with the customer specifying the requirements to be included in each increment.
People not process	The skills of the development team should be recognized and exploited. Team members should be left to develop their own ways of working without prescriptive processes.
Embrace change	Expect the system requirements to change and so design the system to accommodate these changes.
Maintain simplicity	Focus on simplicity in both the software being developed and in the development process. Wherever possible, actively work to eliminate complexity from the system.

•Image source : *Ian Sommerville. 2010. Software Engineering (9th. ed.). Addison-Wesley Publishing Company, USA.*

Where agile methodology not work

Project plan & requirements are clear & unlikely to change

Unclear understanding of Agile Approach among Teams

•Image source : *Ian Sommerville. 2010. Software Engineering (9th. ed.). Addison-Wesley Publishing Company, USA.*

Where agile methodology does not work

Big Enterprises where team collaboration is tough

•Image source : *Ian Sommerville. 2010. Software Engineering (9th. ed.). Addison-Wesley Publishing Company, USA*

Agile Process Models

- Extreme Programming (XP)
- Adaptive Software Development (ASD)
- Dynamic Systems Development Method (DSDM)
- Scrum
- Feature Driven Development (FDD)
- Crystal
- Agile Modelling (AM)

Extreme Programming

- Requirements are expressed as scenarios
- Scenarios are implemented directly as a series of tasks
- Programmers work in pairs
- Develop tests for each task before writing the code
- All tests must be successfully executed when new code is integrated into the system

Extreme Programming

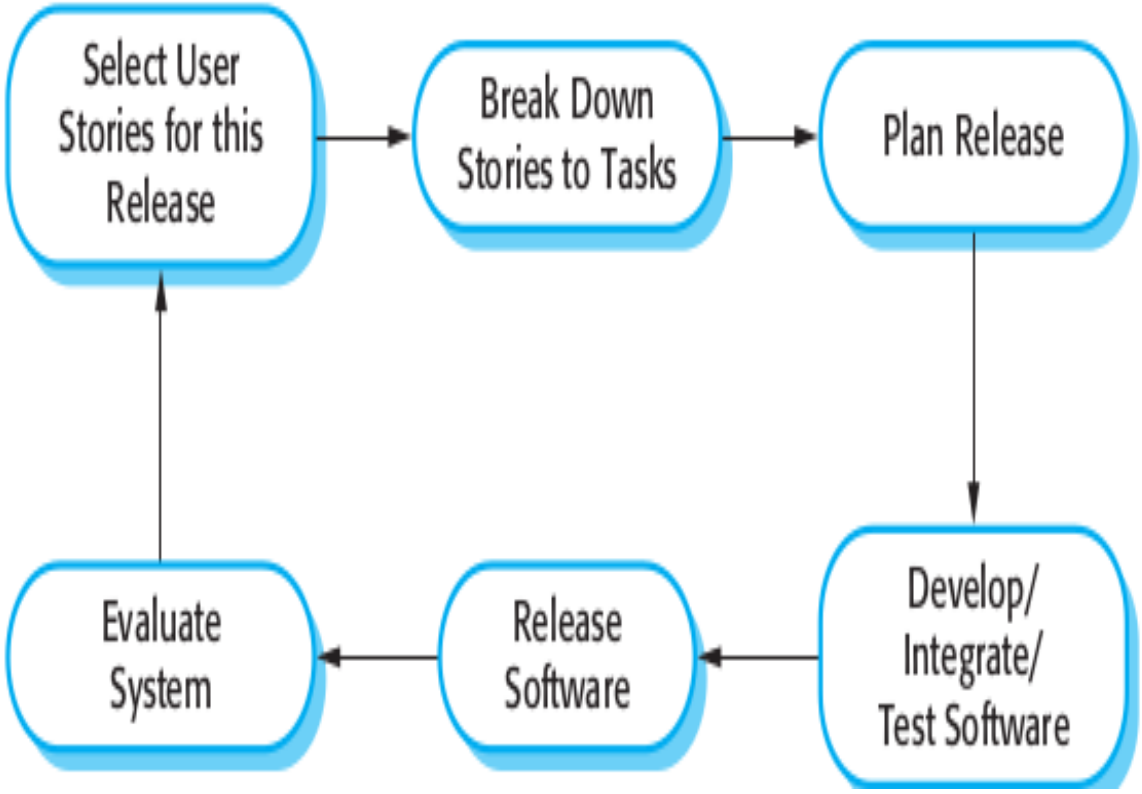


Figure 3.3 The extreme programming release cycle

Principle or practice	Description
Incremental planning	Requirements are recorded on Story Cards and the Stories to be included in a release are determined by the time available and their relative priority. The developers break these Stories into development 'Tasks'. See Figures 3.5 and 3.6.
Small releases	The minimal useful set of functionality that provides business value is developed first. Releases of the system are frequent and incrementally add functionality to the first release.
Simple design	Enough design is carried out to meet the current requirements and no more.
Test-first development	An automated unit test framework is used to write tests for a new piece of functionality before that functionality itself is implemented.
Refactoring	All developers are expected to refactor the code continuously as soon as possible code improvements are found. This keeps the code simple and maintainable.
Pair programming	Developers work in pairs, checking each other's work and providing the support to always do a good job.

•Image source : *Ian Sommerville. 2010. Software Engineering (9th. ed.). Addison-Wesley Publishing Company, USA.*

Collective ownership	The pairs of developers work on all areas of the system, so that no islands of expertise develop and all the developers take responsibility for all of the code. Anyone can change anything.
Continuous integration	As soon as the work on a task is complete, it is integrated into the whole system. After any such integration, all the unit tests in the system must pass.
Sustainable pace	Large amounts of overtime are not considered acceptable as the net effect is often to reduce code quality and medium term productivity
On-site customer	A representative of the end-user of the system (the Customer) should be available full time for the use of the XP team. In an extreme programming process, the customer is a member of the development team and is responsible for bringing system requirements to the team for implementation.

•Image source : *Ian Sommerville. 2010. Software Engineering (9th. ed.). Addison-Wesley Publishing Company, USA.*

Sample story

Prescribing Medication

Kate is a doctor who wishes to prescribe medication for a patient attending a clinic. The patient record is already displayed on her computer so she clicks on the medication field and can select current medication', 'new medication' or 'formulary'.

If she selects 'current medication', the system asks her to check the dose. If she wants to change the dose, she enters the dose and then confirms the prescription.

If she chooses 'new medication', the system assumes that she knows which medication to prescribe. She types the first few letters of the drug name. The system displays a list of possible drugs starting with these letters. She chooses the required medication and the system responds by asking her to check that the medication selected is correct. She enters the dose and then confirms the prescription.

If she chooses 'formulary', the system displays a search box for the approved formulary. She can then search for the drug required. She selects a drug and is asked to check that the medication is correct. She enters the dose and then confirms the prescription.

The system always checks that the dose is within the approved range. If it isn't, Kate is asked to change the dose.

After Kate has confirmed the prescription, it will be displayed for checking. She either clicks 'OK' or 'Change'. If she clicks 'OK', the prescription is recorded on the audit database. If she clicks on 'Change', she reenters the 'Prescribing medication' process.

Figure 3.5 A 'prescribing medication' story.

•Image source : *Ian Sommerville. 2010. Software Engineering (9th. ed.). Addison-Wesley Publishing Company, USA.*

Sample Task cards

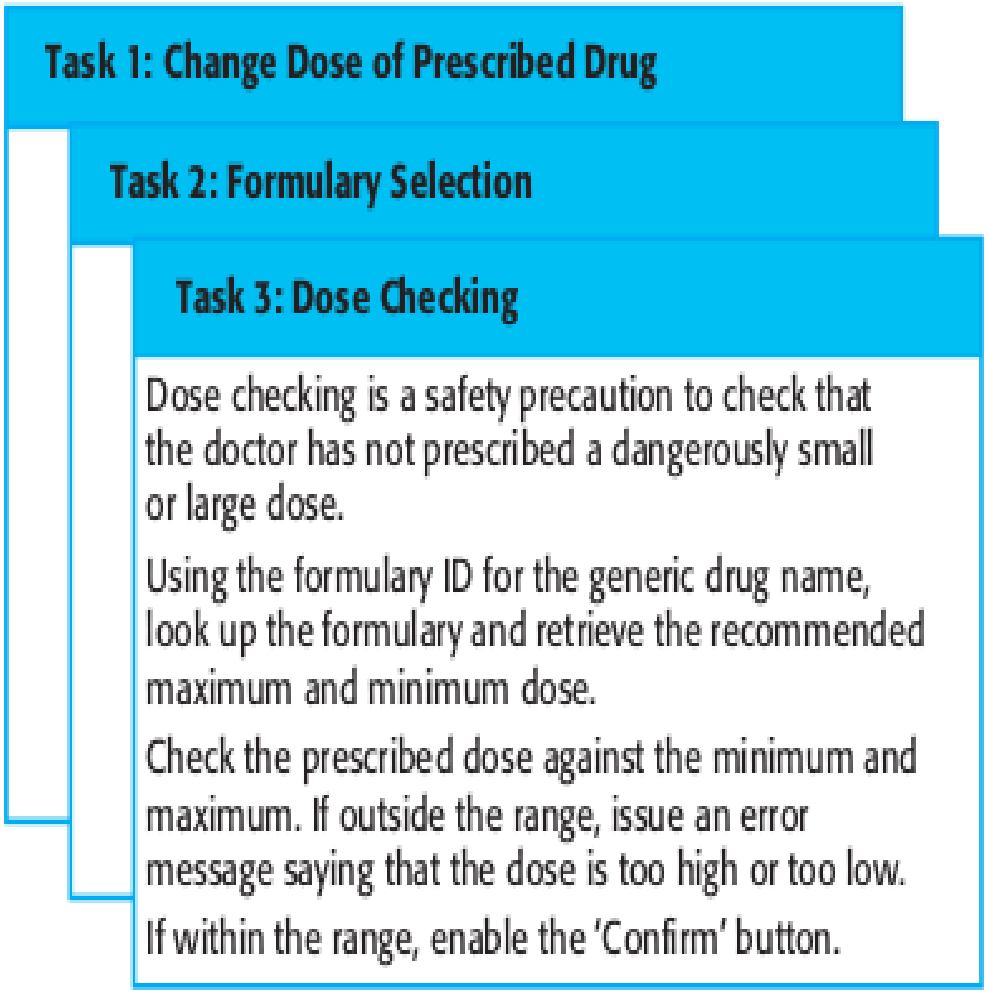


Figure 3.6 Examples of task cards for prescribing medication.

•Image source : *Ian Sommerville. 2010. Software Engineering (9th. ed.). Addison-Wesley Publishing Company, USA.*

Incremental planning, small release, simple design

- Stories will be collected from user
- Tasks will be defined
- Increments of functionality will be provided i.e. incrementally add functionality to the previous release.
- Each increment will be small release with simple design to satisfy current requirements

Test First Development

- The key features of testing in XP are:
- Test-first development
- incremental test development from scenarios
- user involvement in the test development and validation
- Automated testing frameworks Used.

*Image source: https://pixabay.com/vectors/graduation-graduate-student-happy-149646/

Pair Programming

- Observer + Programmer pair
- pairs discuss software before development
- fewer false starts and less rework.
- Less dependency on specific person
- Easy sharing of knowledge
- Reduced risk
- collective ownership
- Informal reviews
- Continuous code Refactoring

Refactoring

- software transformation that improves internal structure
- Preserved external behavior of software i.e. get Maintainability, Extensibility.
- Disciplined way of cleaning code
- Minimize the chances of introducing bugs

Continuous integration

- practice of merging all developers' working copies
- shared mainline is used
- Performed several times a day.
- E.g. GitHub etc

On Site customer

- Alpha testing
- Early system trial by user
- Instant feedback
- Less rework
- Saves time

DIGITAL LEARNING CONTENT

Parul® University

www.paruluniversity.ac.in