Introduction

- Getting started with software engineering

FAQs about software engineering

- What is software?
- What is software engineering?
- What is the difference between software engineering and computer science?
- When do we call software solution to be successful?
- What is a software process or lifecycle?
- What is a software process model?

What is software?

- Computer programs and associated documentation
- Software products may be developed for a particular customer or may be developed for a general market

Software costs

- Software costs often dominate system costs. The costs of software on a PC are often greater than the hardware cost
- Software costs more to maintain than it does to develop. For systems with a long life, maintenance costs may be several times development costs

What is software engineering?

- Software engineering is an engineering discipline which is concerned with all aspects of software production
- Set of rules

What is the difference between software engineering and computer science?

- Computer science is concerned with theory and fundamentals; software engineering is concerned with the practicalities of developing and delivering useful software

Software Systems

- Ubiquitous, used in variety of applications
  - Business, engineering, scientific applications
- Simple to complex, internal to public, single function to enterprise-wide, informational to mission-critical..
- Generic and Customized Software

What are the attributes of good software?

- The software should deliver the required functionality and performance to the user and should be maintainable, dependable and usable
- Maintainability
  - Software must evolve to meet changing needs
- Dependability
  - Software must be trustworthy
- Efficiency
  - Software should not make wasteful use of system resources
- Usability
  - Software must be usable by the users for which it was designed

Successful Software System

- Software development projects have not always been successful
- When do we consider a software application successful?
  - ✔ Development completed
  - ✔ It is useful
  - ✔ It is usable
  - ✔ It is used
- Cost-effectiveness, maintainability implied

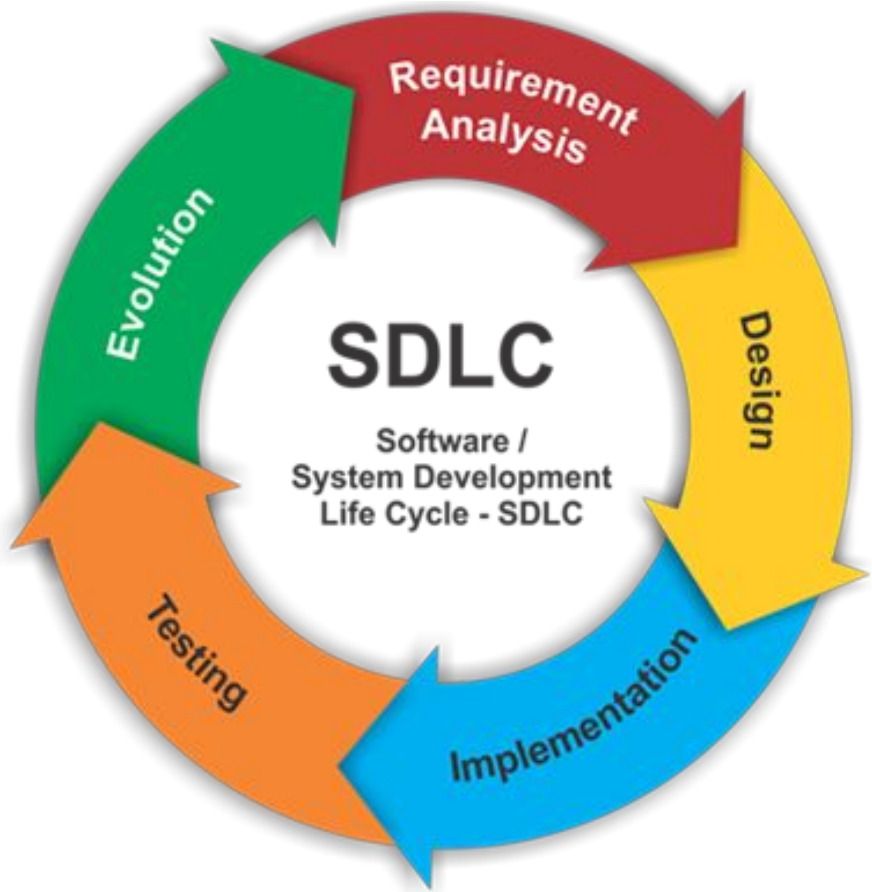
Reason for failure

- Ad hoc software development results in such problems
  - No planning of development work
  - Poor understanding of user requirements
  - No control or review
  - Technical incompetence of developers
  - Poor understanding of cost and effort by both developer and user

What are the key challenges facing software engineering?

- Coping with legacy systems, coping with increasing diversity and coping with demands for reduced delivery times
- Legacy systems
  - Old, valuable systems must be maintained and updated
- Heterogeneity
- Delivery
  - There is increasing pressure for faster delivery of software

Software Development Life Cycle or Software process



Common SDLC models

- Waterfall
- Rapid Prototyping
- Incremental
- Spiral

Software Categories

- Software can be categorized into 6 categories
  - ✔ System Software
  - ✔ Application Software
  - ✔ Engineering Software and Scientific Software
  - ✔ Embedded Software
  - ✔ Web Application
  - ✔ Artificial Intelligence Software

What are the costs of software engineering?

- Roughly 60% of costs are development costs, 40% are testing costs. For custom software, evolution costs often exceed development costs
- Costs vary depending on the type of system being developed and the requirements of system attributes such as performance and system reliability
- Distribution of costs depends on the development model that is used

Characteristics of Good Model

- Should be precisely defined
  - No ambiguity about what is to be done, when, how etc..
- It must be predictable
  - Learn from feedback
  - Can be repeated in other projects with confidence about it's outcome
    - Project-A = done by 3 person in 4 months
    - Project-B = Similar in complexity should also take about same time or with some improvement it should take less time

Advantage of choosing SDLC Model

- Increased development speed
- Increased product quality
- Improved tracking and control
- Improve client relations
- Decreased project risk

No Silver Bullet

- Fredrick Brooks released article Name “No Silver Bullet” on software development
- One of the most famous article in history of software development
- He asserted that there is no single development in either technology or management technique which by itself promise even one order-of-magnitude improvement within a decade in productivity reliability or simplicity

Professional and ethical responsibility

- Software engineering involves wider responsibilities than simply the application of technical skills
- Software engineers must behave in an honest and ethically responsible way if they are to be respected as professionals
- Ethical behaviour is more than simply upholding the law.

Issues of professional responsibility

- *Confidentiality*
  - Engineers should normally respect the confidentiality of their employers or clients irrespective of whether or not a formal confidentiality agreement has been signed.
- *Competence*
  - Engineers should not misrepresent their level of competence. They should not knowingly accept work which is outwith their competence.

Issues of professional responsibility

- *Intellectual property rights*
  - Engineers should be aware of local laws governing the use of intellectual property such as patents, copyright, etc. They should be careful to ensure that the intellectual property of employers and clients is protected.
- *Computer misuse*
  - Software engineers should not use their technical skills to misuse other people's computers. Computer misuse ranges from relatively trivial (game playing on an employer's machine, say) to extremely serious (dissemination of viruses).

ACM/IEEE Code of Ethics

- The professional societies in the US have cooperated to produce a code of ethical practice.
- Members of these organisations sign up to the code of practice when they join.
- The Code contains eight Principles related to the behaviour of and decisions made by professional software engineers, including practitioners, educators, managers, supervisors and policy makers, as well as trainees and students of the profession.

Code of ethics - preamble

- **Preamble**
  - The short version of the code summarizes aspirations at a high level of the abstraction; the clauses that are included in the full version give examples and details of how these aspirations change the way we act as software engineering professionals. Without the aspirations, the details can become legalistic and tedious; without the details, the aspirations can become high sounding but empty; together, the aspirations and the details form a cohesive code.
  - Software engineers shall commit themselves to making the analysis, specification, design, development, testing and maintenance of software a beneficial and respected profession. In accordance with their commitment to the health, safety and welfare of the public, software engineers shall adhere to the following Eight Principles:

Code of ethics - principles

- 1. PUBLIC
  - Software engineers shall act consistently with the public interest.
- 2. CLIENT AND EMPLOYER
  - Software engineers shall act in a manner that is in the best interests of their client and employer consistent with the public interest.
- 3. PRODUCT
  - Software engineers shall ensure that their products and related modifications meet the highest professional standards possible.

Code of ethics - principles

- JUDGMENT
  - Software engineers shall maintain integrity and independence in their professional judgment.
- 5. MANAGEMENT
  - Software engineering managers and leaders shall subscribe to and promote an ethical approach to the management of software development and maintenance.
- 6. PROFESSION
  - Software engineers shall advance the integrity and reputation of the profession consistent with the public interest.

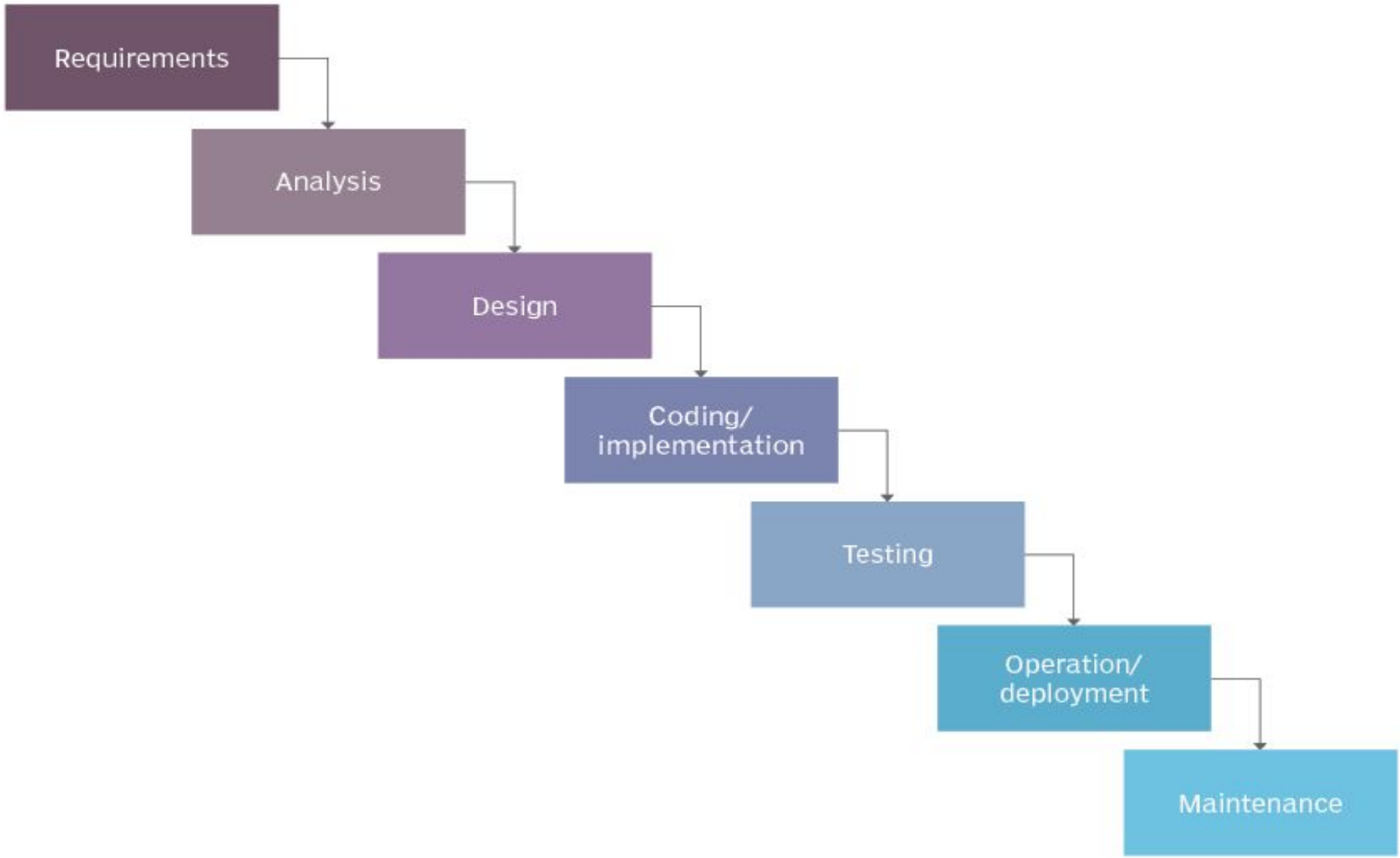
Code of ethics - principles

- 7. COLLEAGUES
  - Software engineers shall be fair to and supportive of their colleagues.
- 8. SELF
  - Software engineers shall participate in lifelong learning regarding the practice of their profession and shall promote an ethical approach to the practice of the profession.

Ethical dilemmas

- Disagreement in principle with the policies of senior management
- Your employer acts in an unethical way and releases a safety-critical system without finishing the testing of the system
- Participation in the development of military weapons systems or nuclear systems

Waterfall model



- Here steps are arranged in linear order
  - A step take inputs from previous step, gives output to next step
  - Exit criteria of a step must match with entry criteria of the succeeding step
- Produces many intermediate deliverable, usually documents
  - Important for quality assurance
- It is widely used when requirements are well understood

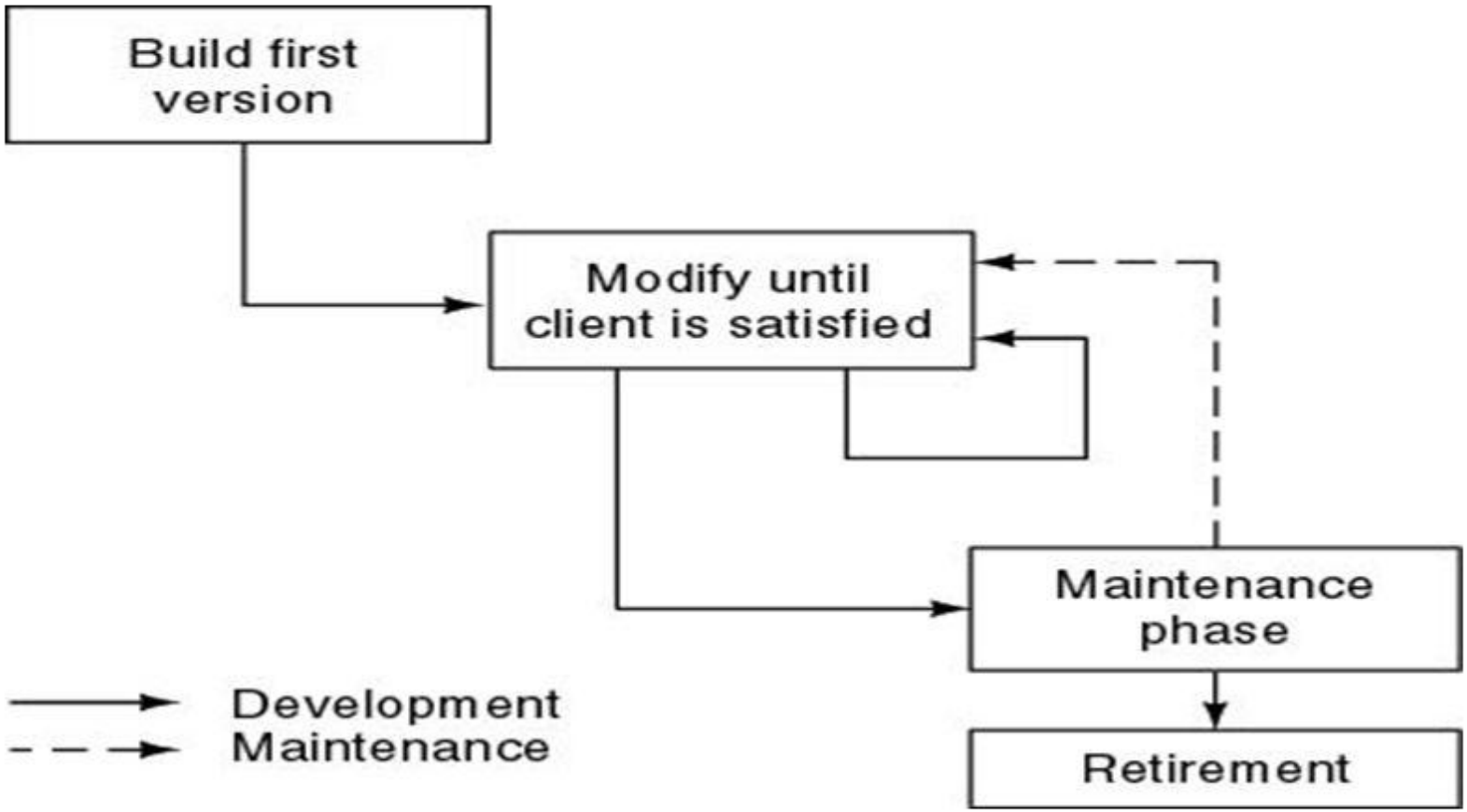
Deliverables in Waterfall model

- Project plan and feasibility report
- Requirements documents
- System design documents
- Test plans and test reports
- Source code
- Software manuals (User manual, installation manual)

Shortcoming of Waterfall model

- Requirements may not be clearly known, especially for applications not having existing manual
- Requirements change with time during project life cycle
  - User may find solution of little use
  - Better to develop in parts in small increments

Build and Fix Model



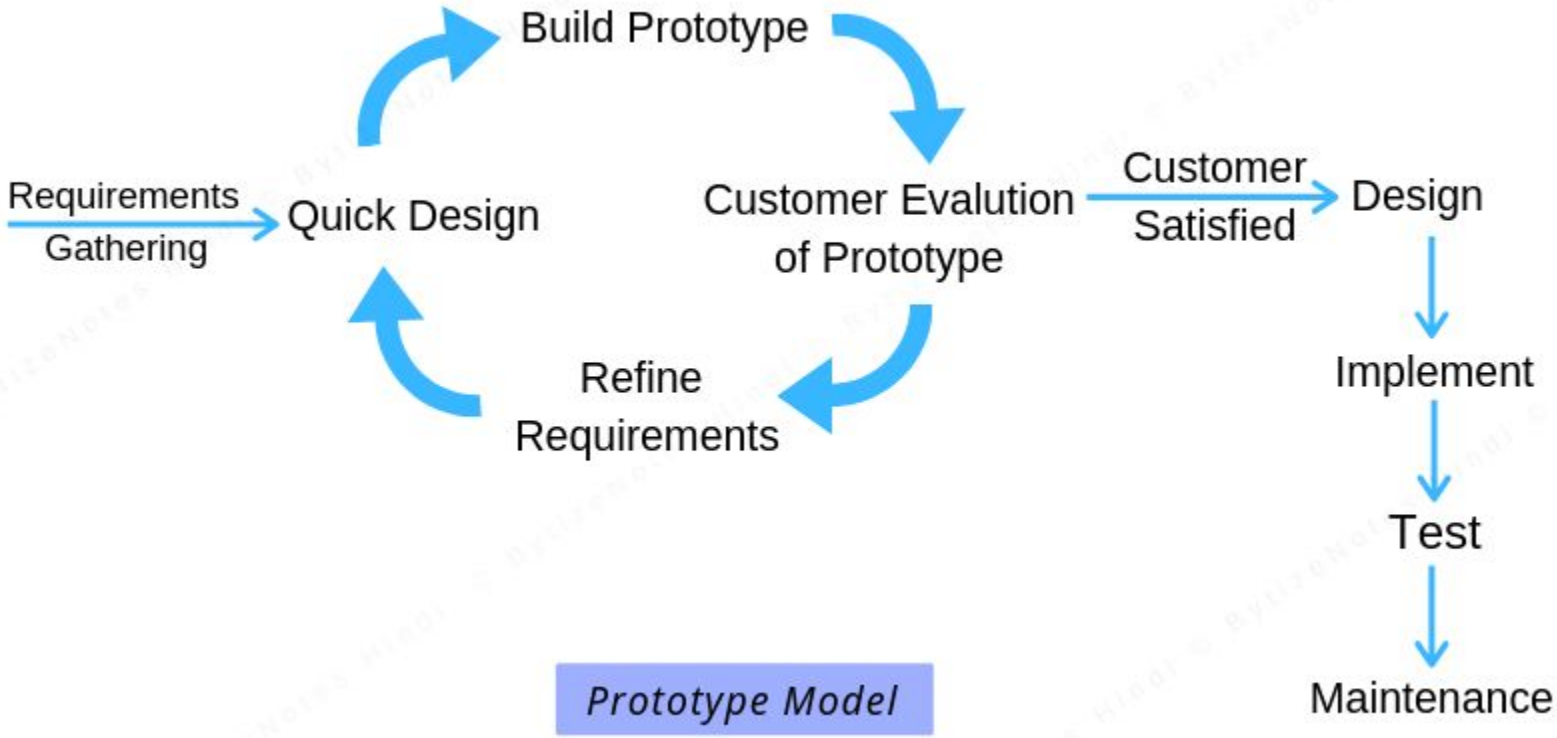
Prototype Model

- When customer or developer is not sure
  - Of requirements
  - Of algorithms, efficiency, human-machine interaction
- A throwaway prototype from currently known user needs
- Working or even paper prototype
- Quick design focuses on aspects visible to user; features clearly understood need not be implemented

Prototype Model

- Prototype is turned to satisfy customer needs
  - Many iterations may be required to incorporate changes and new requirements
- Final product follows usual define-design-build-test life cycle like waterfall mode

Prototype Model

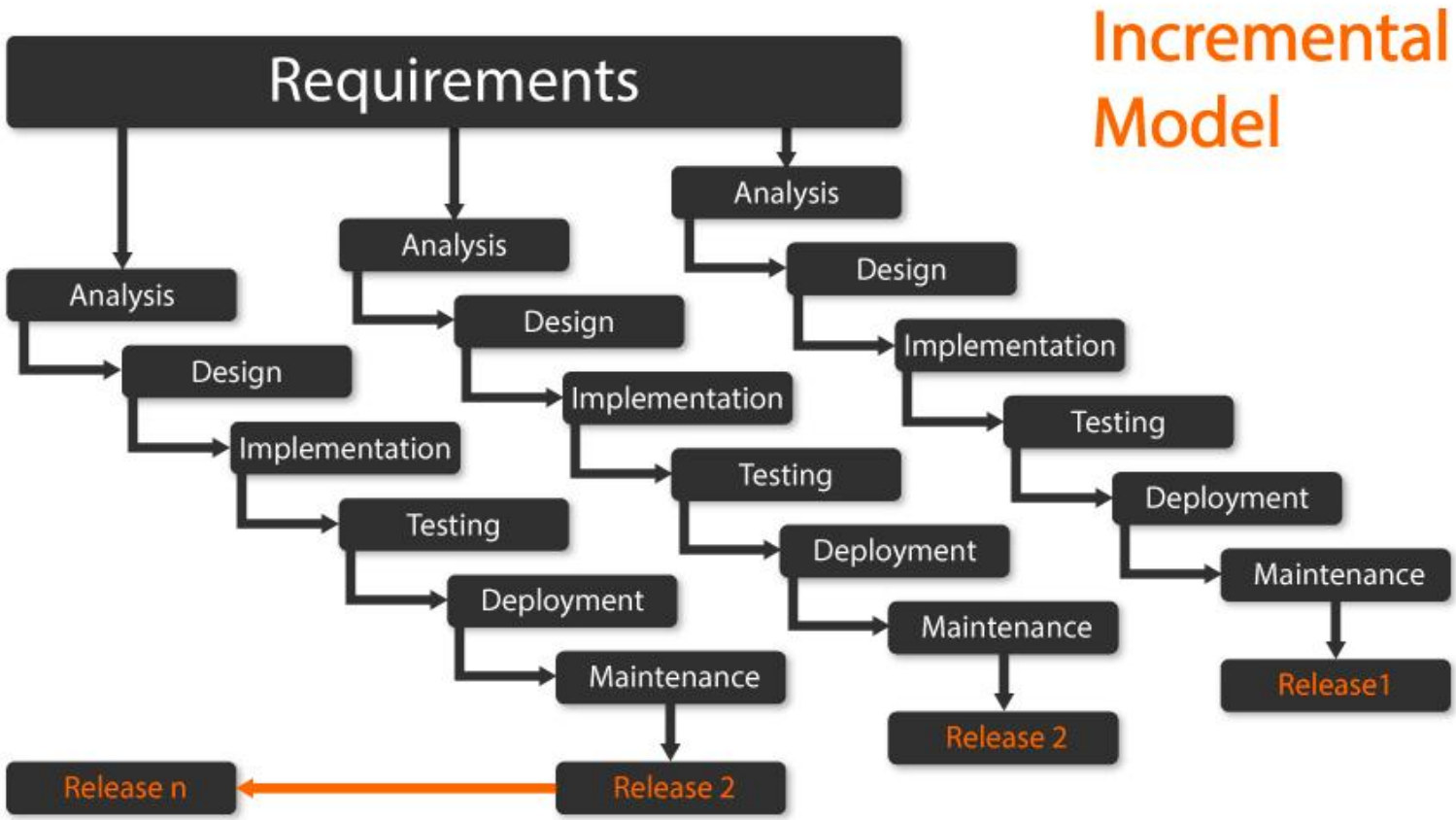


Limitations

- Customer may want prototype itself
- Developer may continue with implementation choices made during prototype
- Good tools need to be acquired for quick development
- May increase project cost due to multiple number of iterations

Incremental Model

- Useful for product development where developers define scope, features to serve many customer
- Early version with limited feature important to establish market and get customer feedback
- A list of features for future versions maintained
- Incremental development reflects the way that we solve problems. We rarely work out complete problem solution in advance but move toward a solution in series of steps


Incremental Model
Requirements
Analysis
Design
Implementation
Testing
Deployment
Maintenance
Release n
Release 2
Release1

Advantages

- Generates working software quickly
- Flexible to customer
- Easier to test and debug
- Easier to manage risk

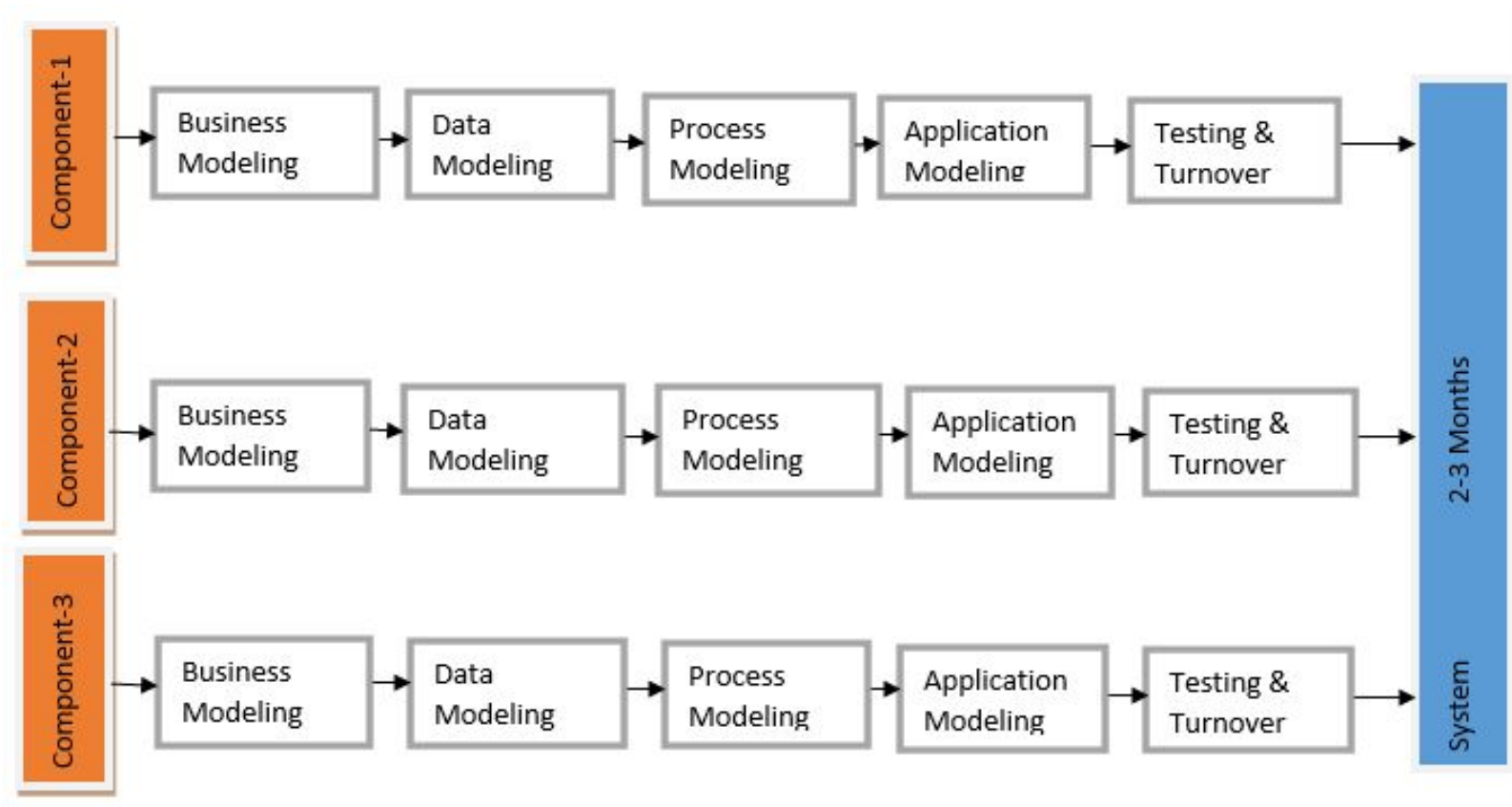
Disadvantages

- Total cost is based on number of iteration
- System structure tends to degrade as new increments are added

RAD

- Extension for the incremental model
- Rapid application development
- The software project which we can brake into modules can use this model
- Development of each module requires basic SDLC steps like waterfall steps

RAD


Component-1
Business Modeling
Data Modeling
Process Modeling
Application Modeling
Testing & Turnover
Component-2
Business Modeling
Data Modeling
Process Modeling
Application Modeling
Testing & Turnover
Component-3
Business Modeling
Data Modeling
Process Modeling
Application Modeling
Testing & Turnover
2-3 Months
System

Five Core Elements

- Business Model
- Data Model
- Process Model
- Application Generation
- Testing and Turnover

Another Explanation of RAD

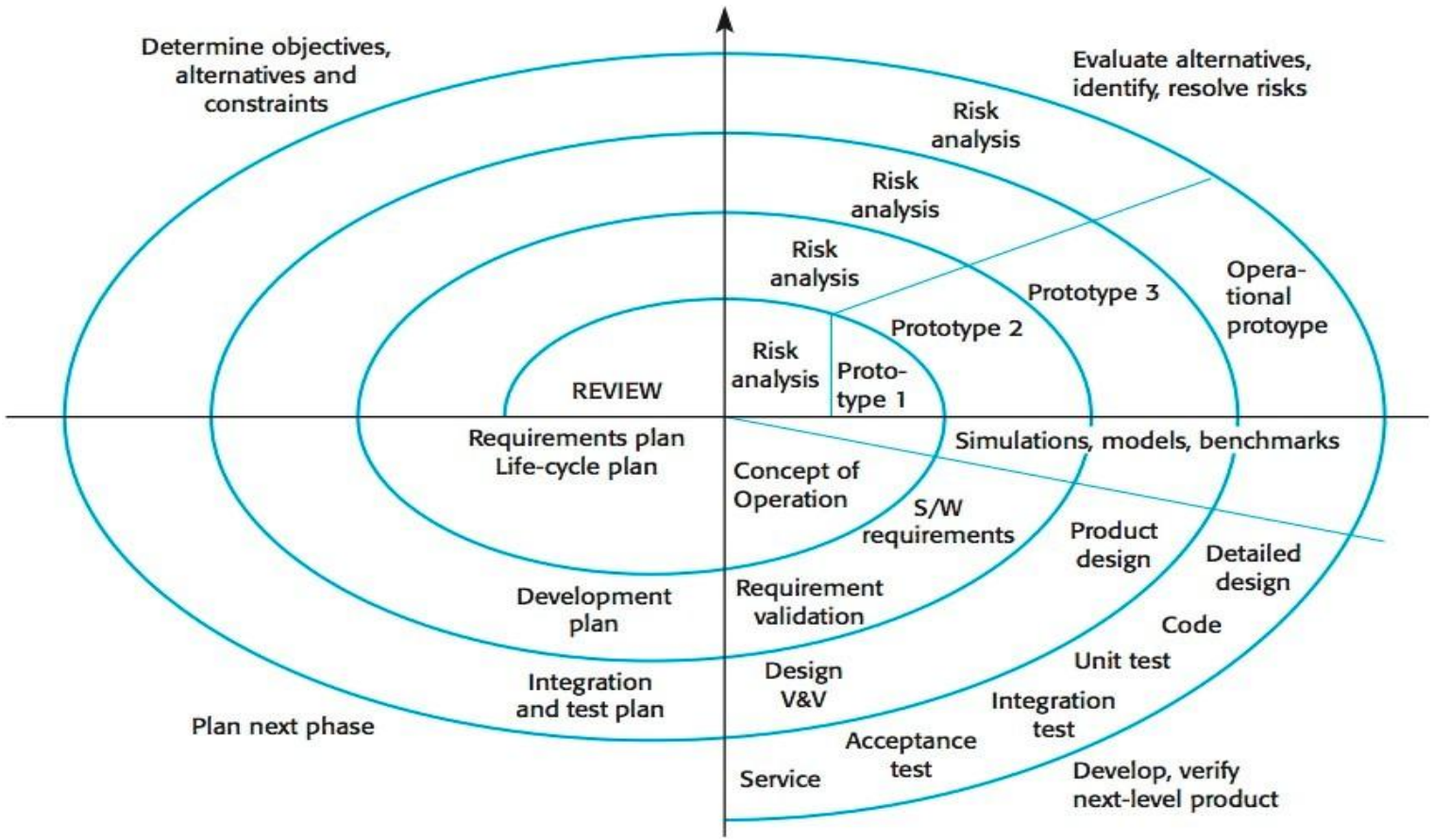
- Requirement Planning
- User Description
- Construction
- Cut-out

Merits and Demerits

- Fast
- Customer satisfaction

- Requires active customer
- Not good for big projects
- Not efficient

Spiral Model



Spiral Model

- Risk driven approach
- Prototyping, simulations, benchmarking may be done to resolve uncertainty/risk
- Development step depends on remaining risk; e.g.
  - Do prototype for user interface risk
  - Use basic waterfall model when user interface and performance issues are understood but only development is remaining

Advantage

- Critical high risk functions are develop first
- The model provides early indication of risk
- User can be closely tide to all lifecycle steps
- Early and frequent feedback
- Frequent releases
- Suitable for large system
- Possible to add additional functionality in later stage

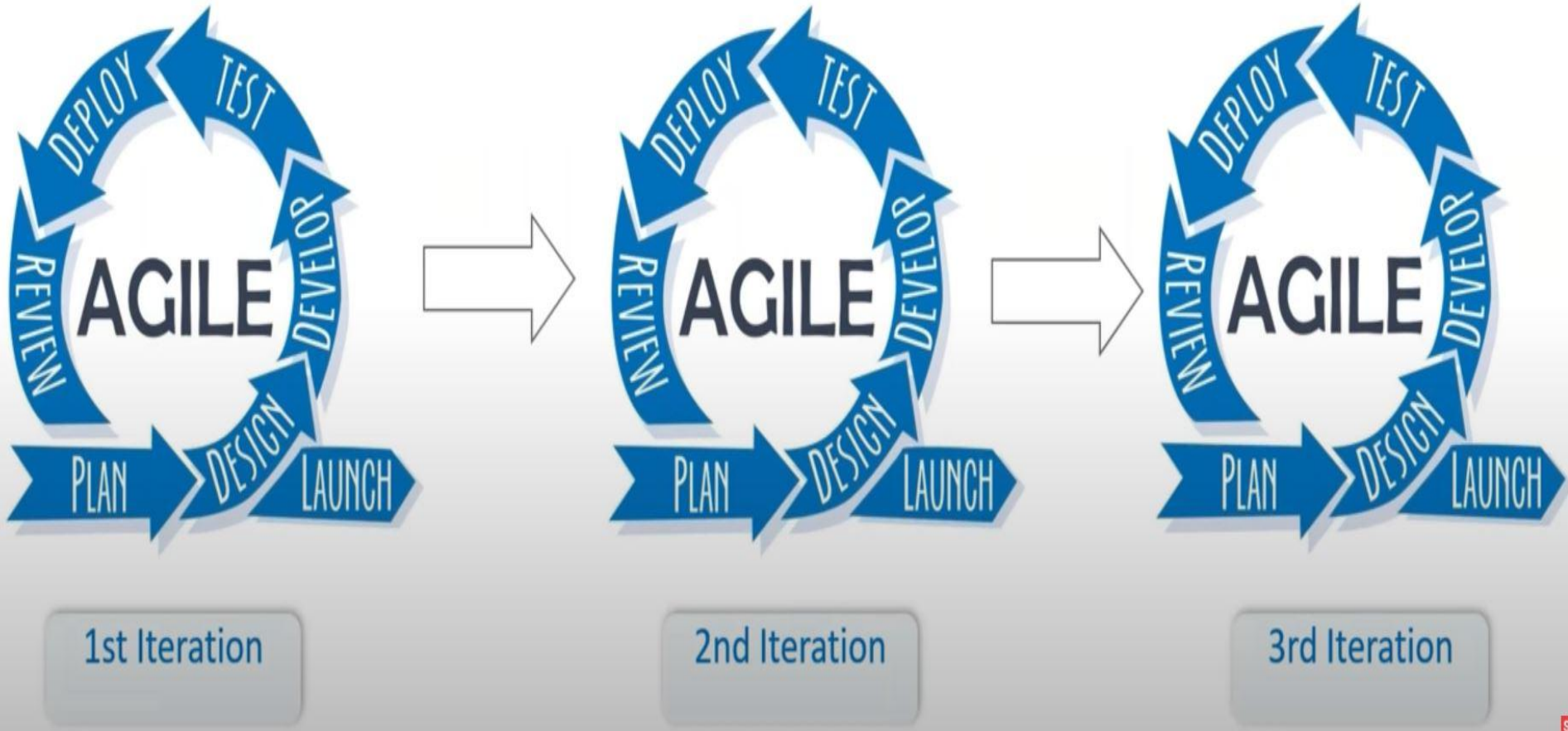
Disadvantage

- What is the impact of using spiral model for small and low risk projects?
  - Time spent for evaluating risks too large for small and low risk projects
  - Time spent for planning, resetting objectives, doing risk analysis and prototyping may be excessive
- Risk is not meeting the schedule on budget
- Expertise are required for risk analysis
- Complex and Costly

Agile

- Main aim of agile is that we build the framework that is nimble enough or agile enough to adjust the changing demands of customer
- It is an iterative and incremental process
- Agile is an idea, based on that idea multiple frameworks are there
  - Scrum, XP, Crystal, FDD, DSDM, Kanban

Agile


DEPLOY
TEST
DEVELOP
REVIEW
AGILE
PLAN
DESIGN
LAUNCH
1st Iteration
2nd Iteration
3rd Iteration

Terms and Value of Agile

- People over Processes and Tools
- Working software over Comprehensive document
- Customer collaboration over Rigid contract
- Responding to change rather than following a plan

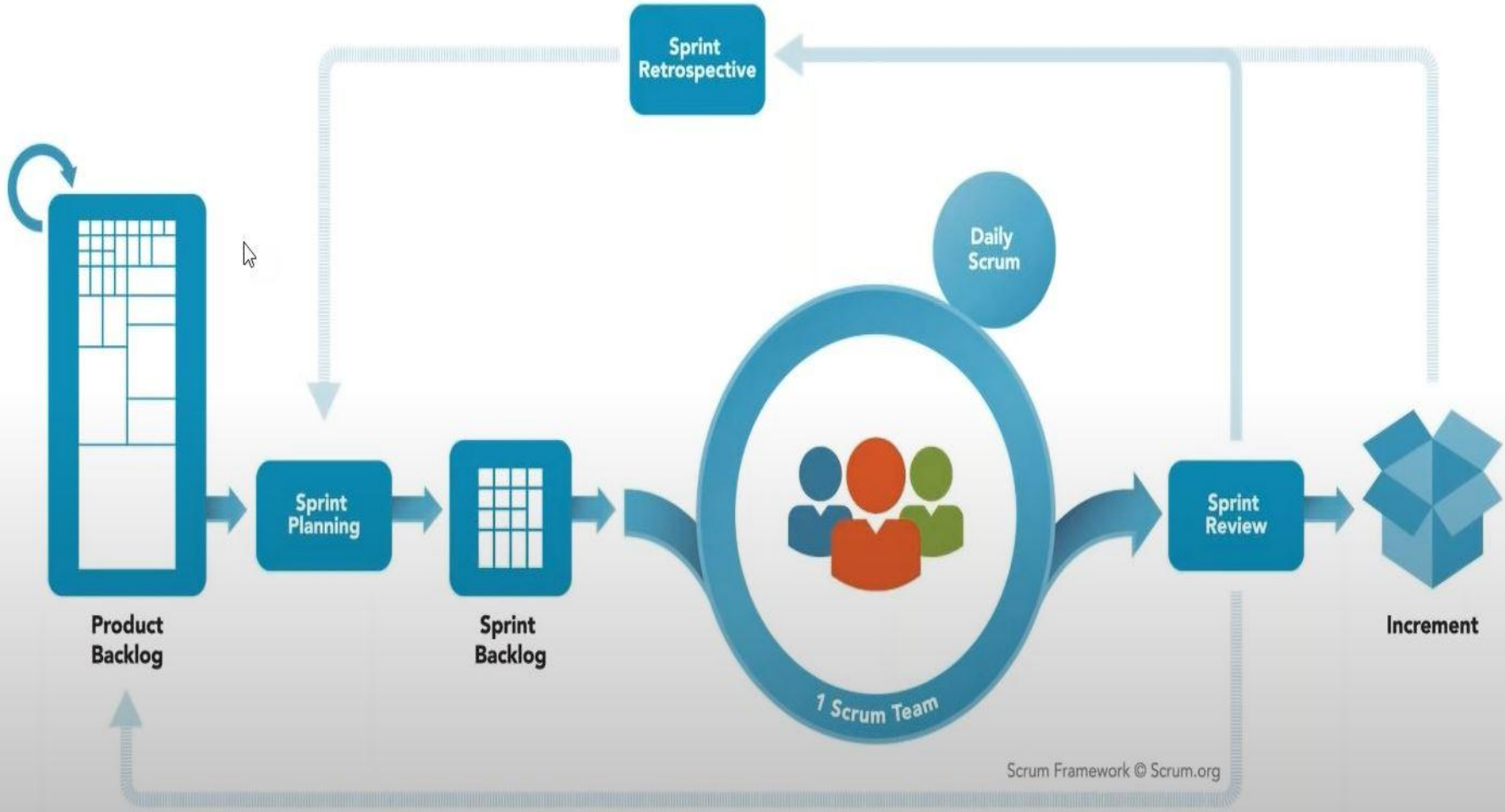
Principle of Agile

- Satisfy the customer
- Welcome changing requirements
- Deliver working software frequently
- Frequent interaction with stockholder
- Motivated individual
- Face-to-face communication
- Measure by working software
- Maintain constant pace
- Sustain technical excellence and good design
- Keep it simple
- Empower self-organization team
- Reflect and adjust continuously

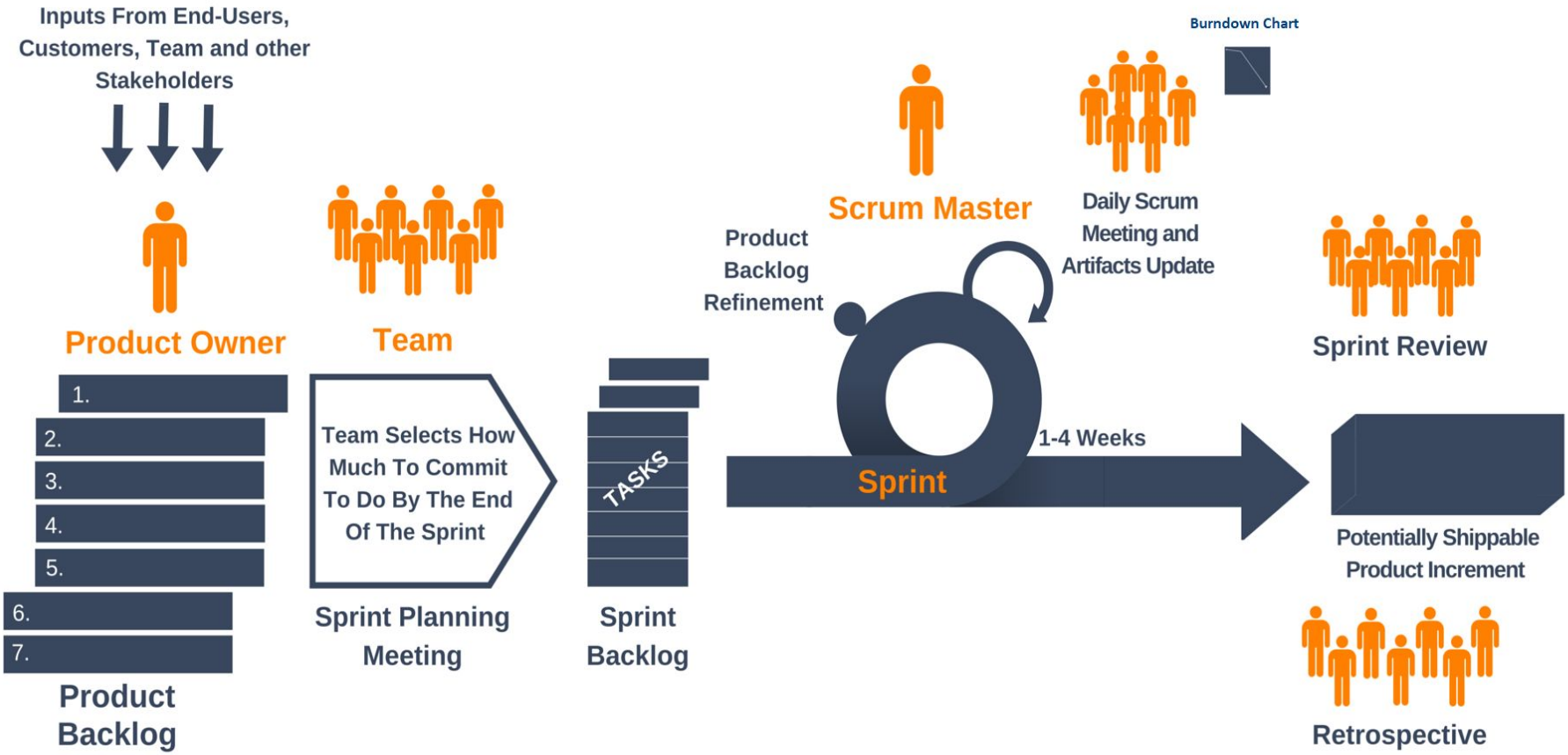
Scrum

- Most popular framework in Agile
- Scrum is a light weight agile project management framework that can be useful to manage creative and incremental projects of all type
- Iterative and Incremental approach
- Inspired by Rugby
- Scrum says we have to have cross-functional team and they have to be focused on advancing the common goal

Scrum Overview



Scrum


Inputs From End-Users, Customers, Team and other Stakeholders
Product Owner
Team
1.
2.
3.
4.
5.
6.
7.
Product Backlog
Team Selects How Much To Commit To Do By The End Of The Sprint
Sprint Planning Meeting
TASKS
Sprint Backlog
Scrum Master
Product Backlog Refinement
Daily Scrum Meeting and Artifacts Update
Burndown Chart
Sprint
1-4 Weeks
Sprint Review
Potentially Shippable Product Increment
Retrospective

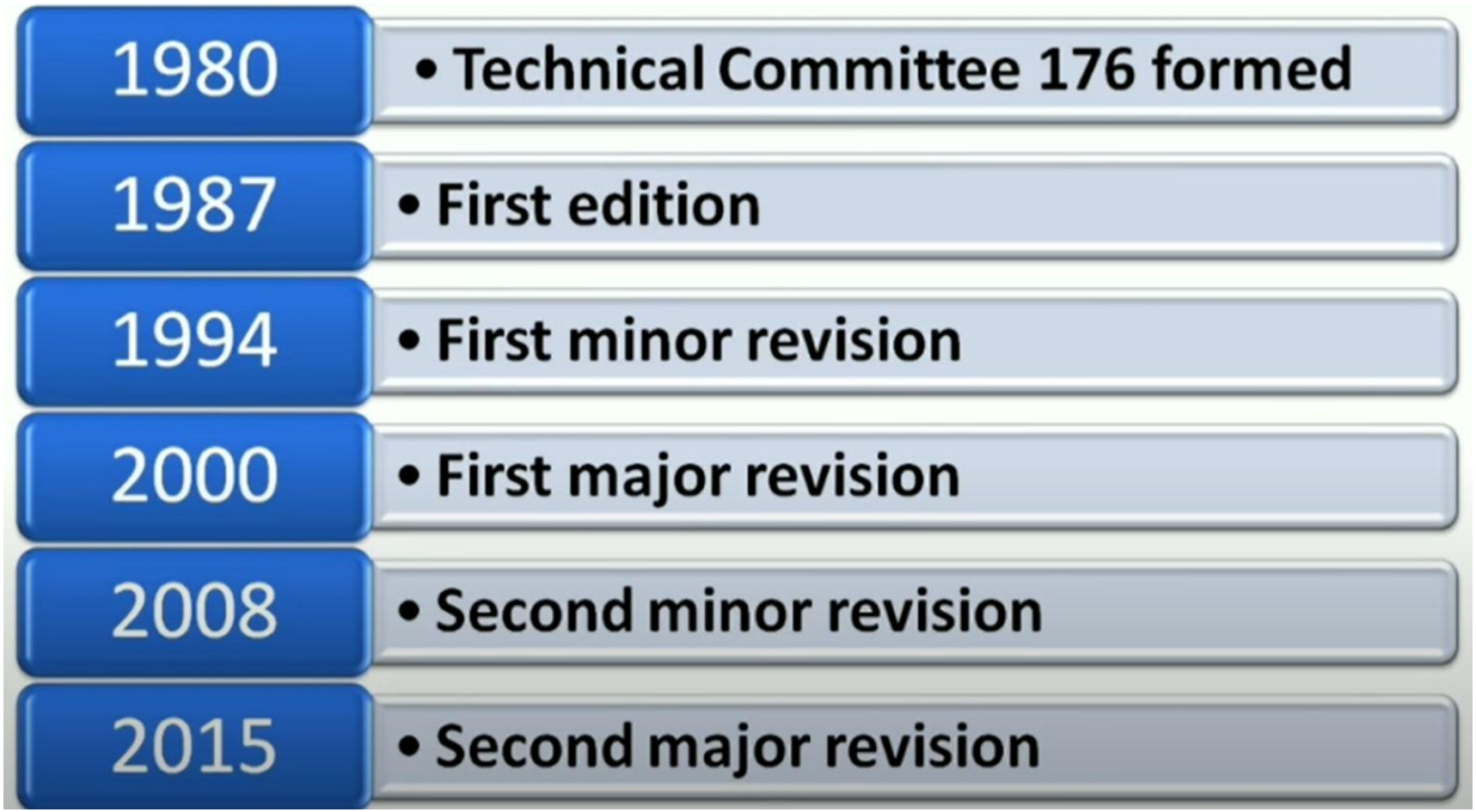
ISO-9000

- ISO stands for International Standardization organization
- It is an international body which is recognized by the UN, which sets up the quality and standards for various production services
- Various certifications are there like
  - ISO 3100 = Risk Management
  - ISO 9000 = Quality Management
  - ISO 14000 = Environment Management
  - ISO 26000 = Social Responsibility

ISO-9000

- ISO 9000 is a family of quality based standards
- Quality Management certification category
- What is Quality Management ?
- Consist 4 family member
  - ISO : 9001
  - ISO : 9002
  - ISO : 9003
  - ISO : 9004

ISO-9000 Timeline



ISO-9000 Quality Principles

- Customer Focus
- Effective Leadership
- Involvement
- Process Approach
- System Approach to Management
- Continual Improvement
- Factual Approach to Decision Making
- Mutually Beneficial Supplier Relationship

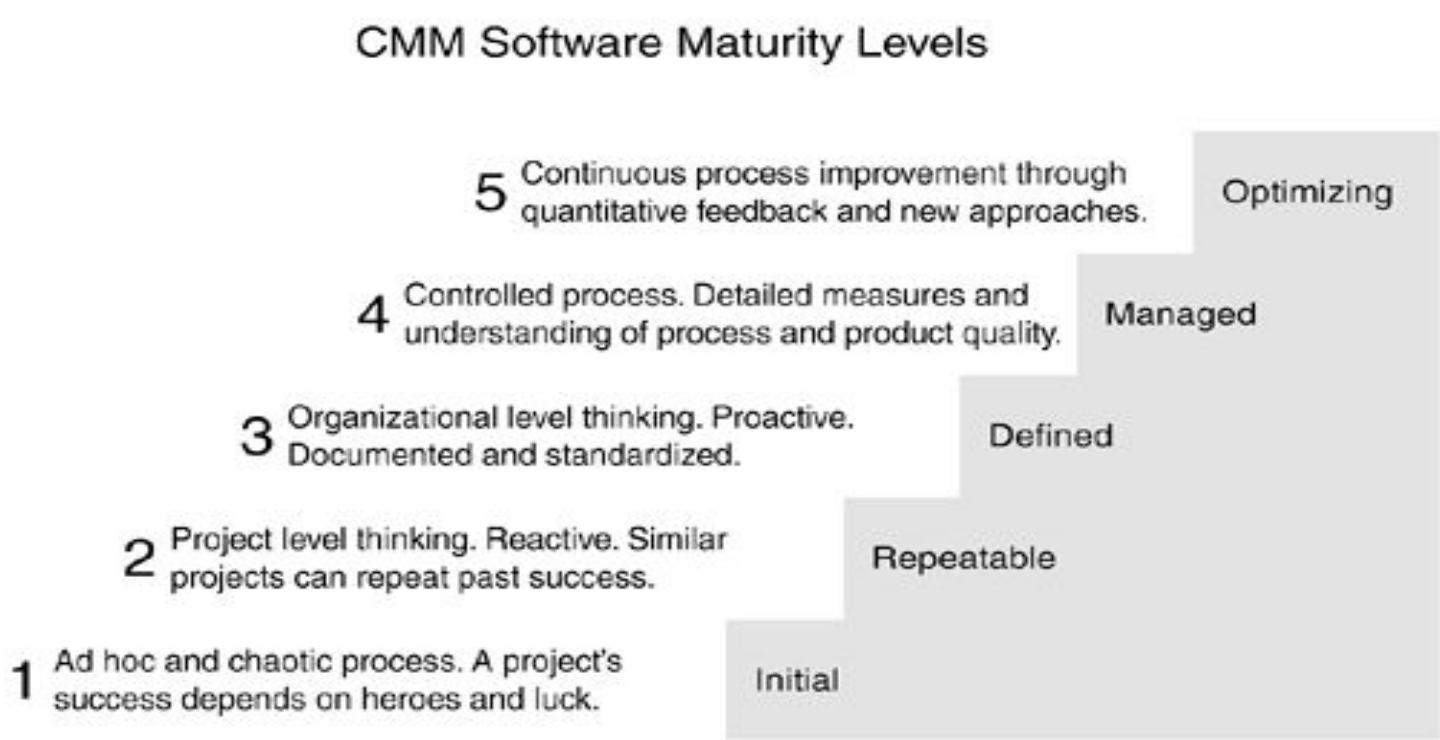
CMM

- Capability Maturity Model
- Developed by SEI (Software Engineering Institute)
- CMM is a strategy for improving the software process, irrespective of the actual life cycle model, which is being used
- CMM is used to judge the maturity of software processes of an organization, and to identify key practices that are required to increase the maturity of these processes

CMM Levels

- There are 5 levels in CMM
- Measures a maturity level of an organization based on certain key process area
  - The project
  - Clients
- Each level ranks the organization according to what kind of standard process company following for the particular subject area
- One is minimum, 5 is maximum

CMM Levels


CMM Software Maturity Levels
5 Continuous process improvement through quantitative feedback and new approaches.
Optimizing
4 Controlled process. Detailed measures and understanding of process and product quality.
Managed
3 Organizational level thinking. Proactive. Documented and standardized.
Defined
2 Project level thinking. Reactive. Similar projects can repeat past success.
Repeatable
1 Ad hoc and chaotic process. A project's success depends on heroes and luck.
Initial

ISO 9000	SEICMM
ISO 9000 is a set of international standarads on quality management and quality assurance developed to help companies effectively document the quality system elements needed to an efficient quality system.	SEI (Software Engineering Institute), Capability Maturity Model (CMM) specifies an increasing series of levels of a software development organization.
Focus is customer supplier relationship, attempting to reduce customer's risk in choosing a supplier.	Focus on the software supplier to improve its interval processes to achieve a higher quality product for the benefit of the customer.
It is created for hard goods manufacturing industries.	It is created for software industry.
ISO9000 is recognized and accepted in most of the countries.	SEICMM is used in USA, less widely elsewhere.
It specifies concepts, principles and safeguards that should be in place.	CMM provides detailed and specific definition of what is required for given levels.
This establishes one acceptance level.	It assesses on 5 levels.
Its certification is valid for three years.	It has no limit on certification.
It focuses on inwardly processes.	It focus outwardly.
It has no level.	It has 5 levels: (a). Initial (b). Repeatable (c). Defined (d). Managed (e). Optimized
It is basically an audit.	It is basically an appraisal.
It is open to multi sector.	It is open to IT/ITES.
Follow set of standards to make success repeatable.	It emphasizes a process of continuous improvement.