MCSE 544

Software Design and Integration

Software Classification And Development Models

Dr Md Mehedi Hasan
Stamford University Bangladesh

Introduction

Classification

❑ Based on application:

1. **System Software –**
   System Software is necessary to manage the computer resources and support the execution of application programs. Software like operating systems, compilers, editors and drivers etc., come under this category. A computer cannot function without the presence of these.
2. **Networking and Web Applications Software –**
   Networking Software provides the required support necessary for computers to interact with each other and with data storage facilities. The networking software is also used when software is running on a network of computers (such as World Wide Web). It includes all network management software, server software, security and encryption software and software to develop web-based applications like HTML, PHP, XML, etc.
3. **Embedded Software –**
   This type of software is embedded into the hardware normally in the Read Only Memory (ROM) as a part of a large system and is used to support certain functionality under the control conditions. Examples are software used in instrumentation and control applications like washing machines, satellites, microwaves etc.

Classification

4. **Reservation Software –**
   A Reservation system is primarily used to store and retrieve information and perform transactions related to air travel, car rental, hotels, or other activities. They also provide access to bus and railway reservations, although these are not always integrated with the main system.
5. **Business Software –**
   This category of software is used to support the business applications and is the most widely used category of software. Examples are software for inventory management, accounts, banking, hospitals, schools, stock markets, etc.
6. **Entertainment Software –**
   Education and entertainment software provides a powerful tool for educational agencies, especially those that deal with educating young children. There is a wide range of entertainment software such as computer games, educational games, translation software, mapping software, etc.
7. **Artificial Intelligence Software –**
   Software like expert systems, decision support systems, pattern recognition software, artificial neural networks, etc. come under this category. They involve complex problems which are not affected by complex computations using non-numerical algorithms.

Classification

8. **Scientific Software –**
   Scientific and engineering software satisfies the needs of a scientific or engineering user to perform enterprise specific tasks. Such software is written for specific applications using principles, techniques and formulae specific to that field. Examples are software like MATLAB, AUTOCAD, PSPICE, ORCAD, etc.
9. **Utilities Software –**
   The programs coming under this category perform specific tasks and are different from other software in terms of size, cost and complexity. Examples are anti-virus software, voice recognition software, compression programs, etc.
10. **Document Management Software –**
    A Document Management Software is used to track, manage and store documents in order to reduce the paperwork. Such systems are capable of keeping a record of the various versions created and modified by different users (history tracking). They commonly provide storage, versioning, metadata, security, as well as indexing and retrieval capabilities.

Classification

- **Based on copyright:**

•Commercial –
It represents the majority of software which we purchase from software companies, commercial computer stores, etc. In this case, when a user buys a software, they acquire a license key to use it.

•Shareware –
Shareware software is also covered under copyright, but the purchasers are allowed to make and distribute copies with the condition that after testing the software, if the purchaser adopts it for use, then they must pay for it.

•Freeware –
In general, according to freeware software licenses, copies of the software can be made both for archival and distribution purposes but here, distribution cannot be for making a profit.

•Public Domain –
In case of public domain software, the original copyright holder explicitly relinquishes all rights to the software. Hence software copies can be made both for archival and distribution purposes with no restrictions on distribution.

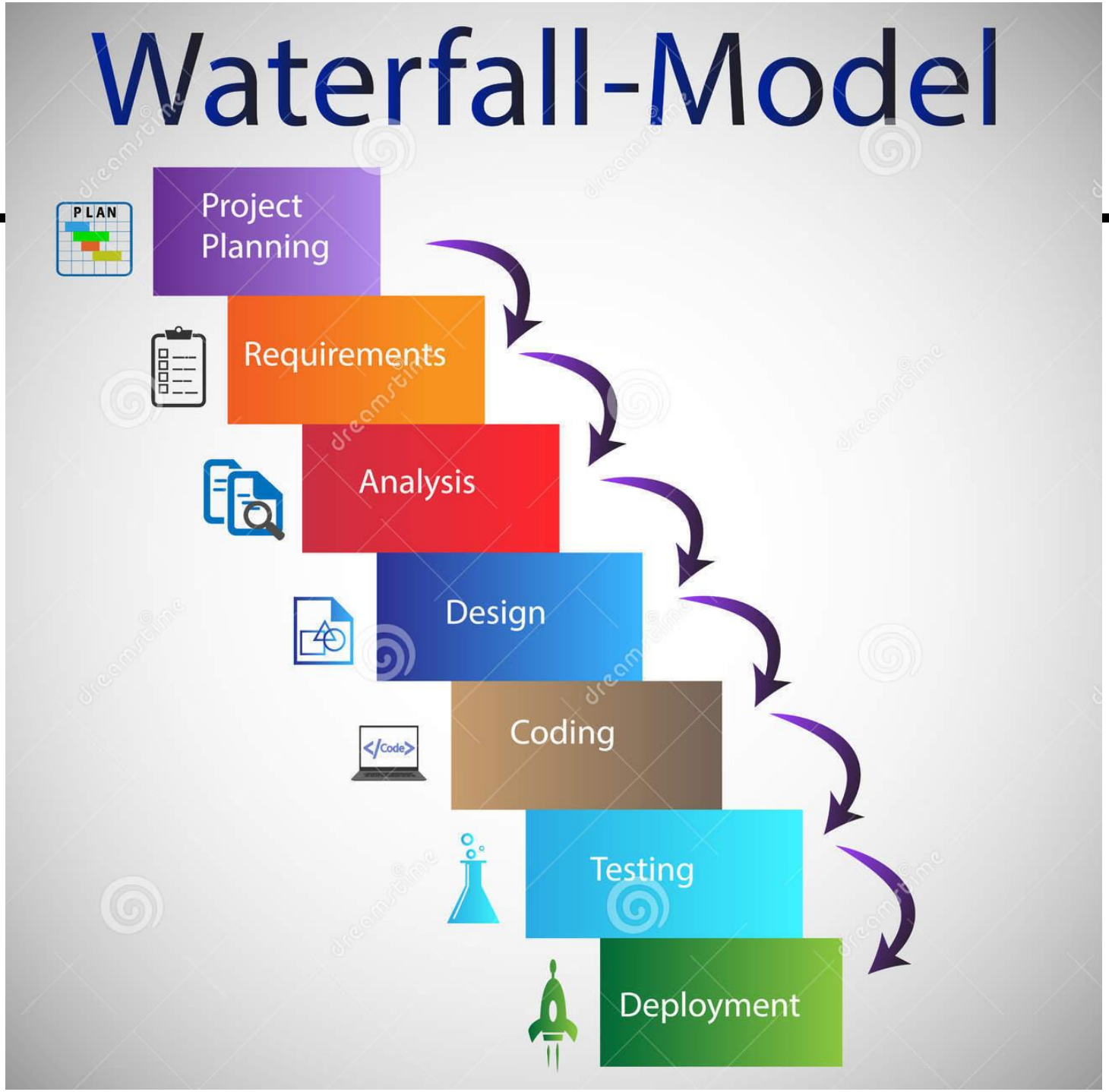
Different Types of Software Development Model

List of Software Development Model

There are many software development models and methods are available in the market. All of this models are mentioned below:

1. Waterfall model
2. V model
3. Incremental model
4. RAD model
5. Agile model
6. Iterative model
7. Spiral model

Waterfall-Model



Waterfall Model

Advantages of Waterfall Model

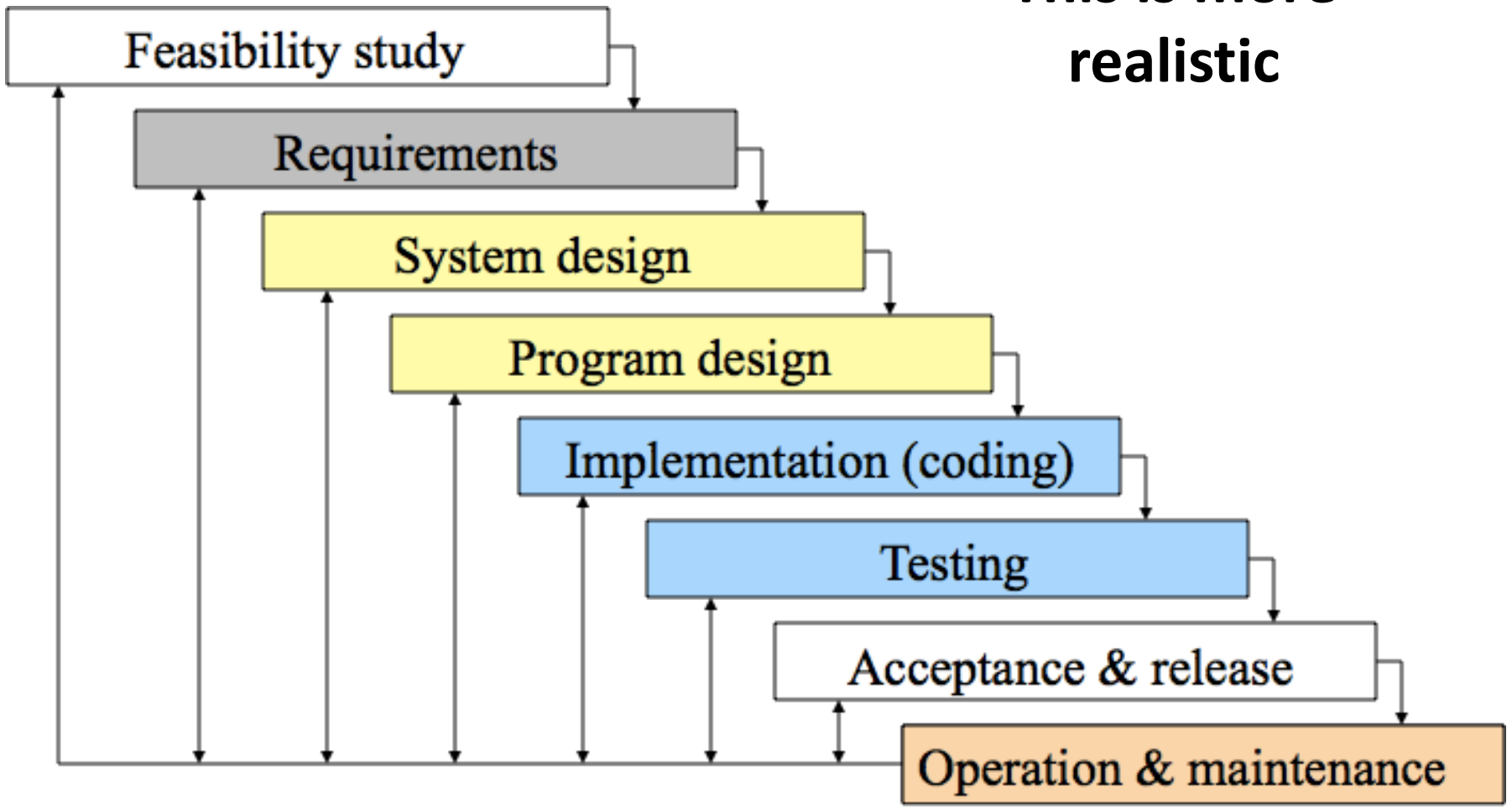
- ✓ Simple and easy to understand and use.
- ✓ Easy to manage due to the rigidity of the model – each phase has specific deliverables and a review process.
- ✓ Phases are processed and completed one at a time.
- ✓ Works well for smaller projects where requirements are very well understood.

Disadvantages of Waterfall Model

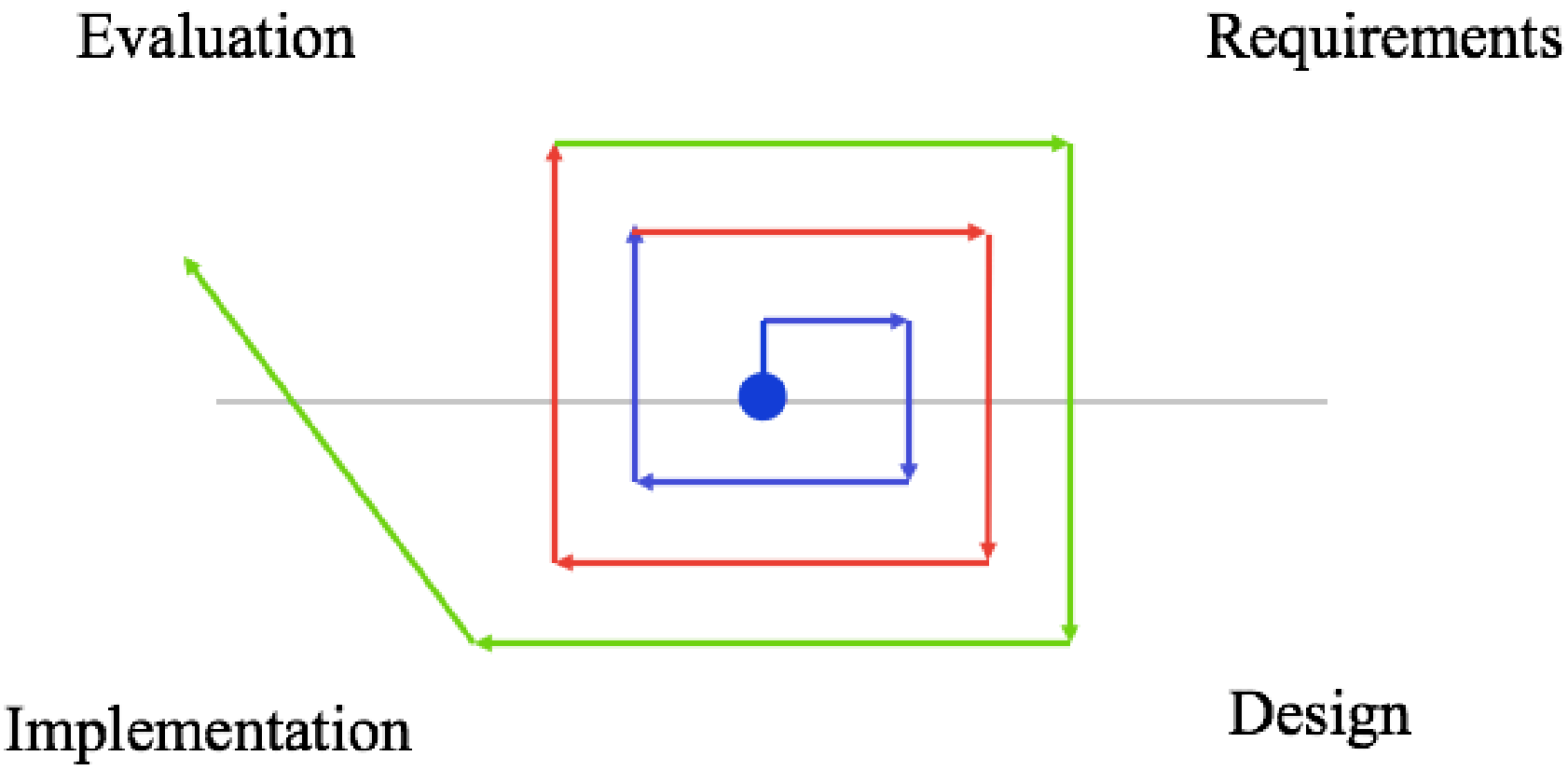
- ✓ Once an application is in the testing stage, it is very difficult to go back and change something that was not well-thought out in the concept stage.
- ✓ No working software is produced until late during the life cycle.
- ✓ High amounts of risk and uncertainty.
- ✓ Not a good model for complex and object-oriented projects.
- ✓ Poor model for long and ongoing projects.
- ✓ Not suitable for the projects where requirements are at a moderate to high risk of changing.

Modified Waterfall

This is more realistic



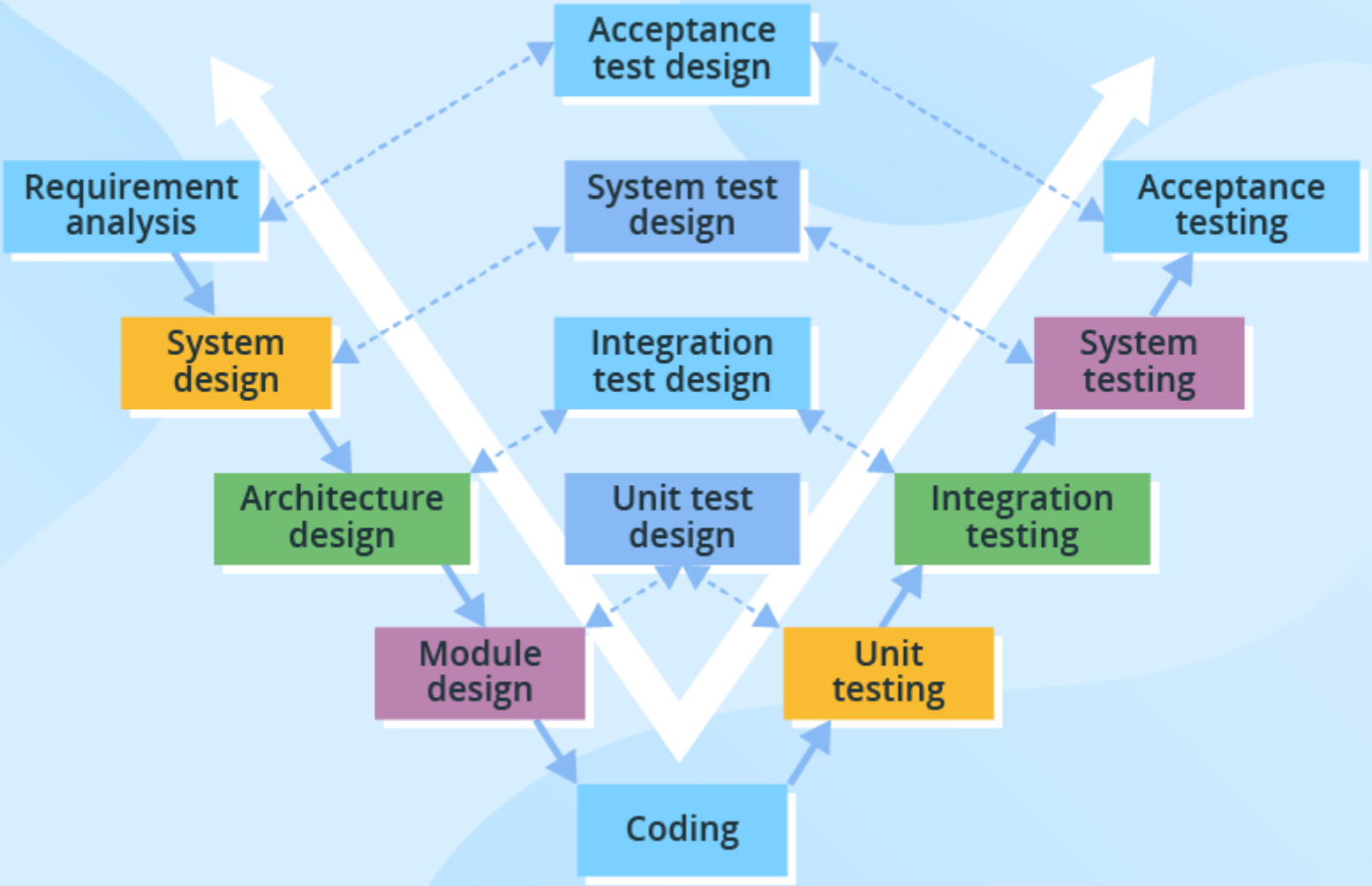
Iterative Refinement



Iterative Refinement Discussion

- Pros
  - Complete (bare-bones) system done quickly
  - Can correct mistakes in early design stages
- Cons
  - Throw away a lot of code
  - Can encourage feature bloat
  - Can lead to half-done features

V-MODEL


Acceptance test design
Requirement analysis
System test design
Acceptance testing
System design
Integration test design
System testing
Architecture design
Unit test design
Integration testing
Module design
Unit testing
Coding

V Model

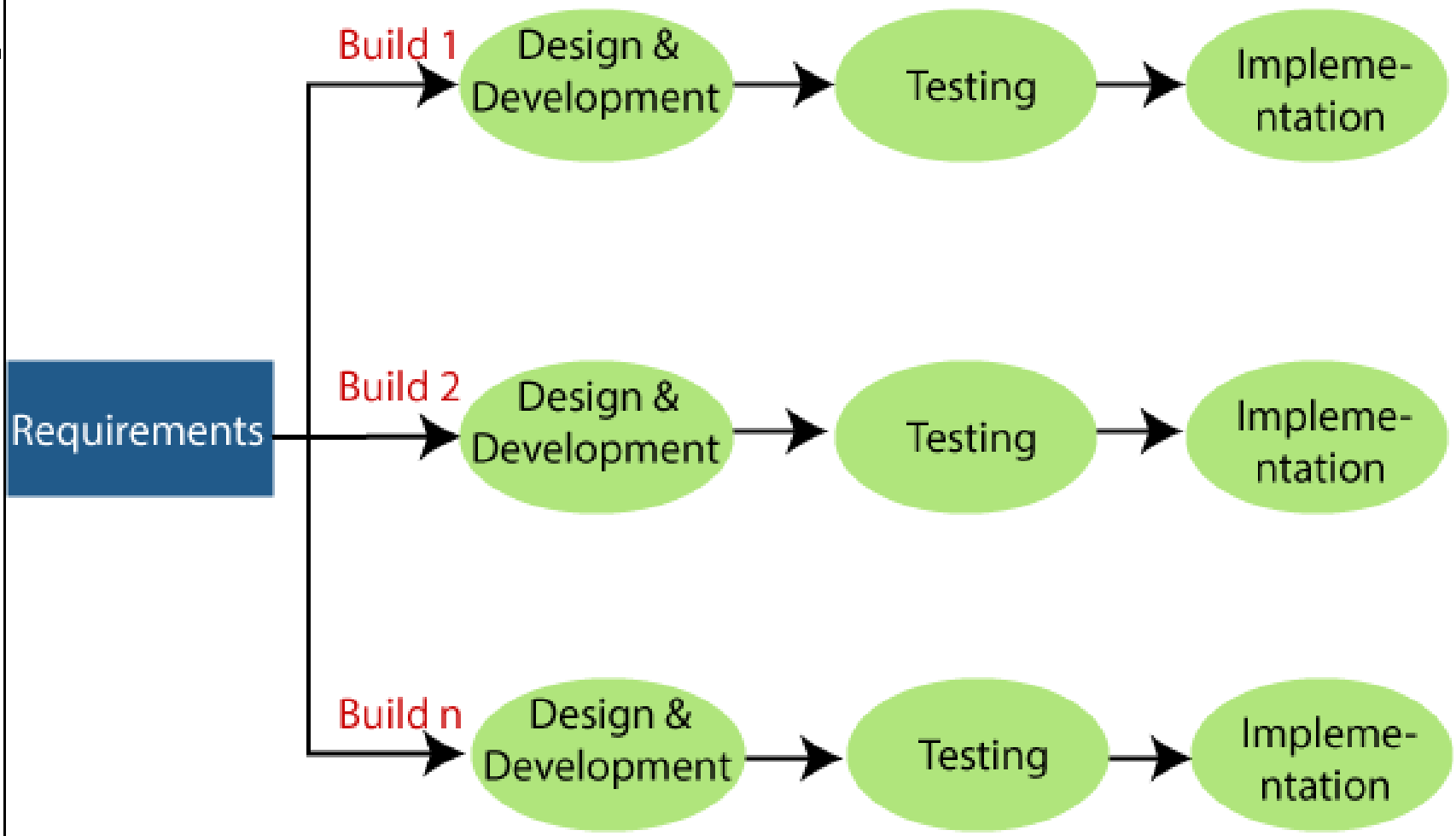
Advantages of V Model

- ✓ Simple and easy to use.
- ✓ Testing activities like planning, test designing happens well before coding. This saves a lot of time. Hence higher chance of success over the waterfall model.
- ✓ Proactive defect tracking – that is defects are found at an early stage.
- ✓ Avoids the downward flow of the defects.
- ✓ Works well for small projects where requirements are easily understood.

Disadvantages of V Model

- ✓ Very rigid and less flexible.
- ✓ Software is developed during the implementation phase, so no early prototypes of the software are produced.
- ✓ If any changes happen in midway, then the test documents along with required documents has to be updated.

Incremental Model



Incremental Model

Advantages of Incremental Model

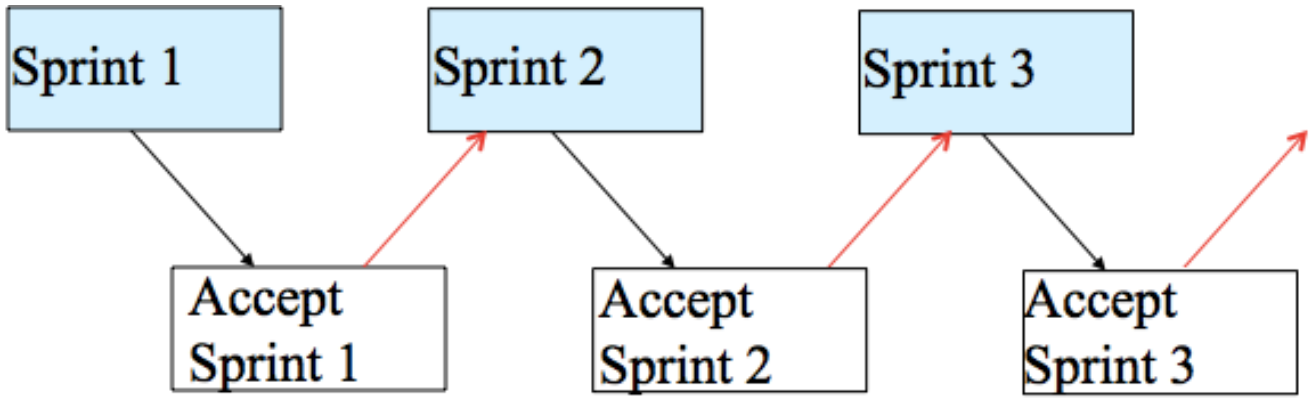
- ✓ Generates working software quickly and early during the software life cycle.
- ✓ More flexible – less costly to change scope and requirements.
- ✓ Easier to test and debug during a smaller iteration.
- ✓ The customer can respond to each bolt.
- ✓ Lowers initial delivery cost.
- ✓ Easier to manage risk because risky pieces are identified and handled during it'd iteration.

Disadvantages of Incremental Model

- ✓ Needs good planning and design.
- ✓ Needs a clear and complete definition of the whole system before it can be broken down and built incrementally.
- ✓ Total cost is higher than a waterfall.

Incremental

- In each increment (sprint) the team works through the full software development cycle and ends up with new production-ready features
- Each sprint is assigned a fixed (and short) time frame, e.g. 4 weeks
- Team size involved in a sprint is usually small (5-10)



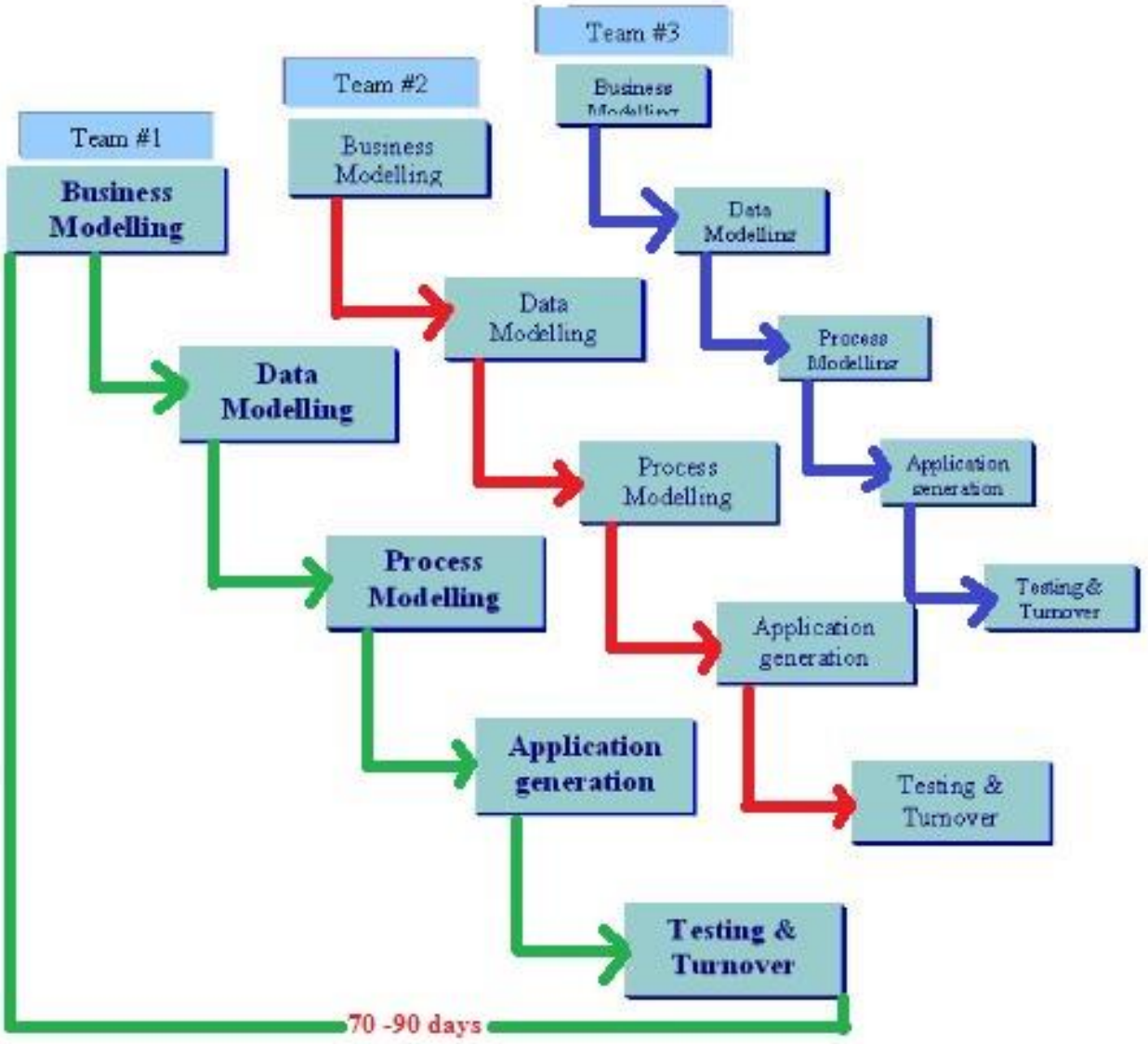


Fig:- RAD (Rapid Application Development) Model

RAD Model

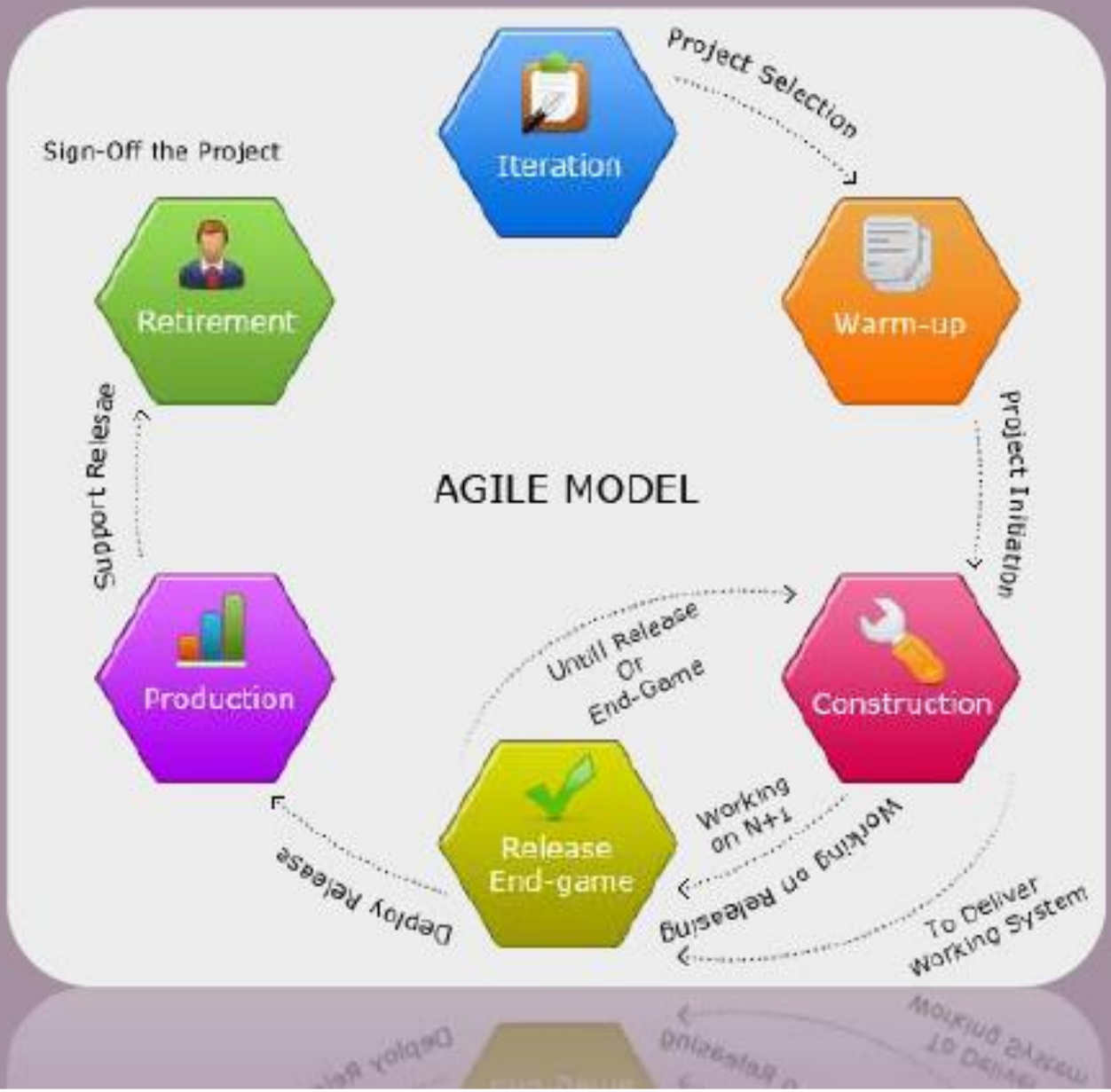
Advantages of RAD Model

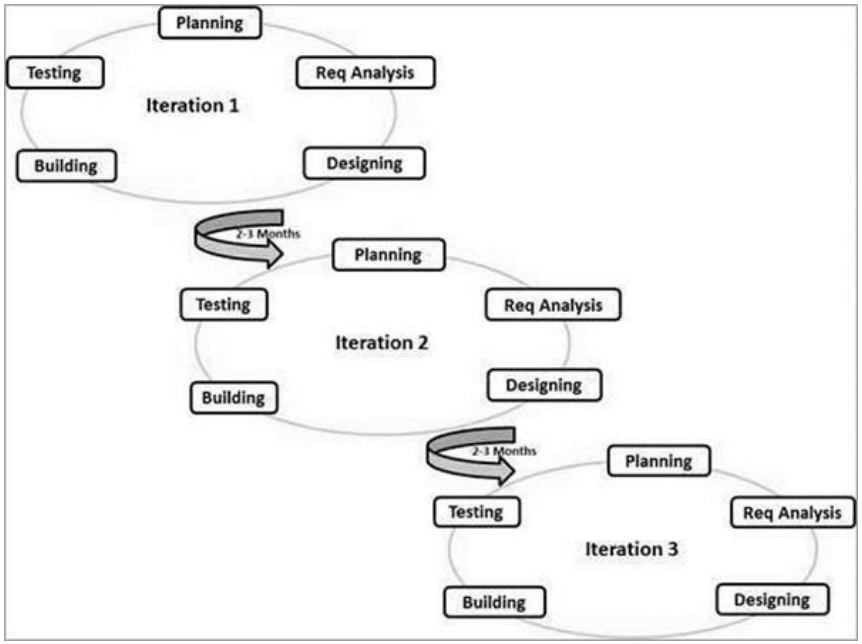
- ✓ Reduced development time.
- ✓ Increases reusability of components
- ✓ Quick initial reviews occur
- ✓ Encourages customer feedback
- ✓ Integration from very beginning solves a lot of integration issues.

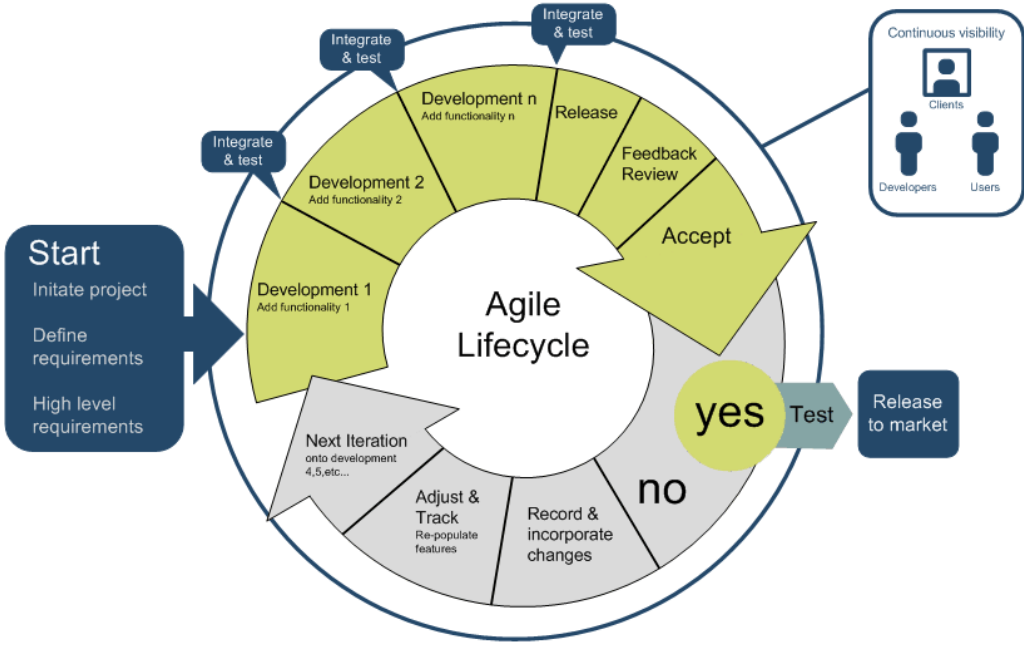
Disadvantages of RAD Model

- ✓ Depends on strong team and individual performances for identifying business requirements.
- ✓ Only systems that can be modularized can be built using RAD
- ✓ Requires highly skilled developers/designers.
- ✓ High dependency on modeling skills

Agile Model


Iteration
Project Selection
Warm-up
Project Initiation
Construction
Working on Releasing
To Deliver Working System
Release End-game
Deploy Release
Production
Support Release
Retirement
Sign-Off the Project
AGILE MODEL
Until Release Or End-Game
Working on N+1


Planning
Testing
Req Analysis
Iteration 1
Building
Designing
2-3 Months
Planning
Testing
Req Analysis
Iteration 2
Building
Designing
2-3 Months
Planning
Testing
Req Analysis
Iteration 3
Building
Designing


Start
Initate project
Define requirements
High level requirements
Integrate & test
Integrate & test
Integrate & test
Development 1
Add functionality 1
Development 2
Add functionality 2
Development n
Add functionality n
Release
Feedback Review
Accept
Agile Lifecycle
yes
Test
Release to market
no
Record & incorporate changes
Adjust & Track
Re-populate features
Next Iteration
onto development 4,5,etc...
Continuous visibility
Clients
Developers
Users

Agile Model

Advantages of Agile Model

- ✓ Customer satisfaction by rapid, continuous delivery of useful software.
- ✓ People and interactions are emphasized rather than process and tools.

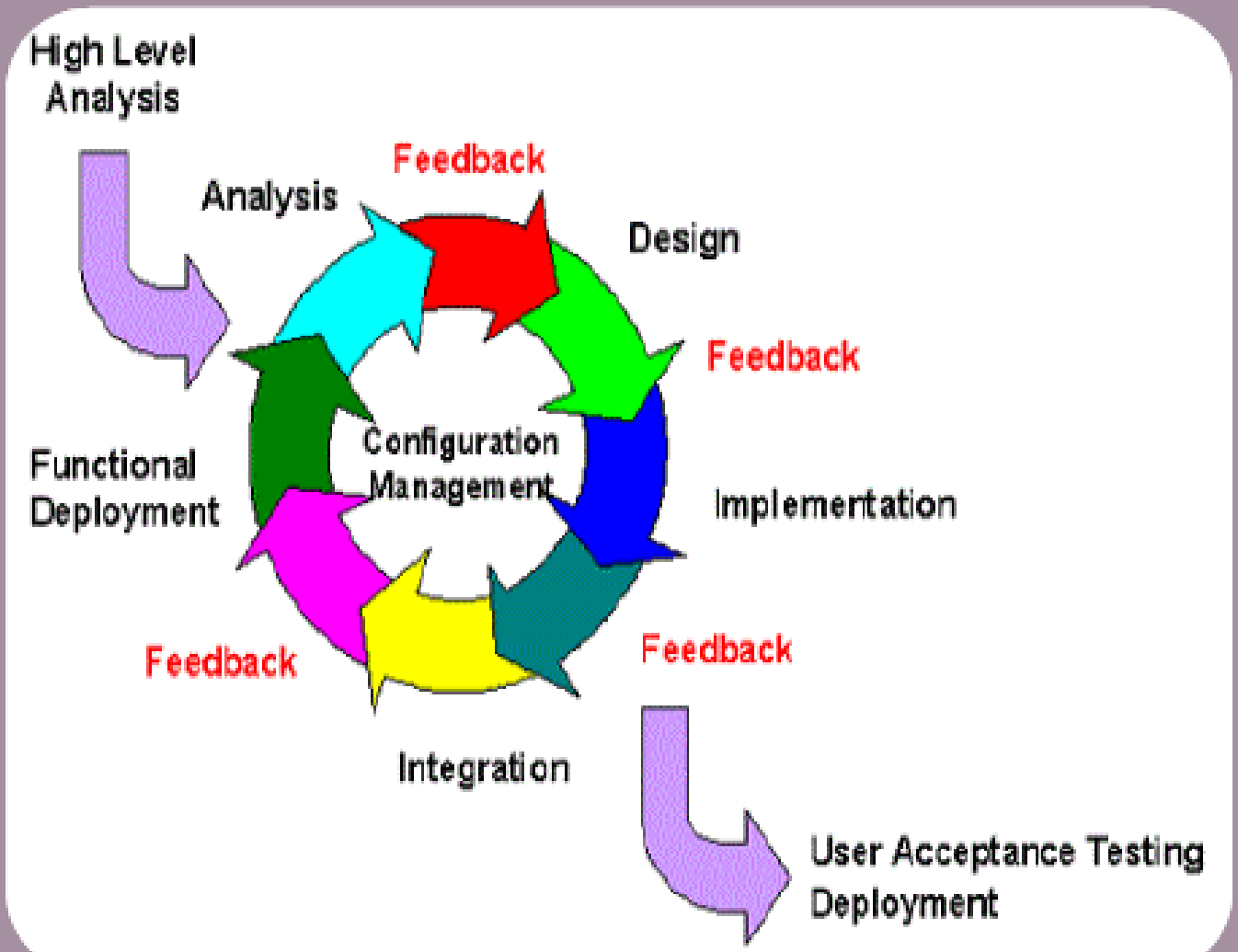
Working software is delivered frequently (weeks rather than months).

- ✓ Close, daily cooperation between business people and developers.
- ✓ Continuous attention to technical excellence and good design.
- ✓ Regular adaptation to changing circumstances.
- ✓ Even late changes in requirements are welcomed

Disadvantages of Agile Model

- ✓ In case of some software deliverables, especially the large ones, it is difficult to assess the effort required at the beginning of the software development life cycle.
- ✓ There is a lack of emphasis on necessary designing and documentation.
- ✓ The project can easily get taken off track if the customer representative is not clear what the final outcome that they want.

Iterative Model



Iterative Model

Advantages of Iterative Model

✓ In the iterative model we are building and improving the product step by step. Hence we can track the defects at early stages. This avoids the downward flow of the defects.

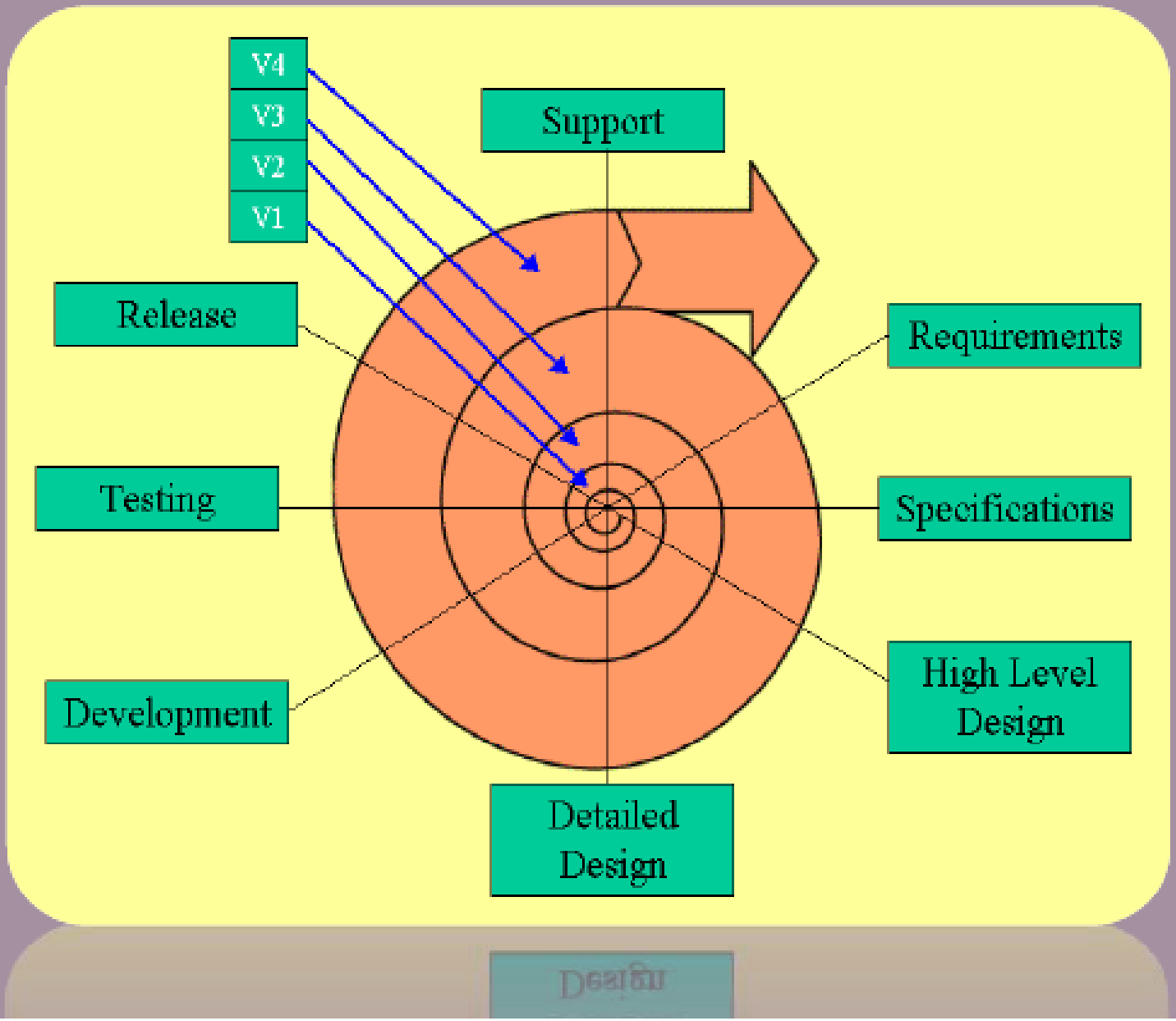
✓In the iterative model we can get the reliable user feedback. When presenting sketches and blueprints of the product to users for their feedback, we are effectively asking them to imagine how the product will work.

Disadvantages of Iterative Model

✓ Each phase of an iteration is rigid with no overlaps

✓ Costly system architecture or design issues may arise because not all requirements are gathered up front for the entire Lifecycle.

Spiral Model



Spiral Model

Advantages of Spiral Model

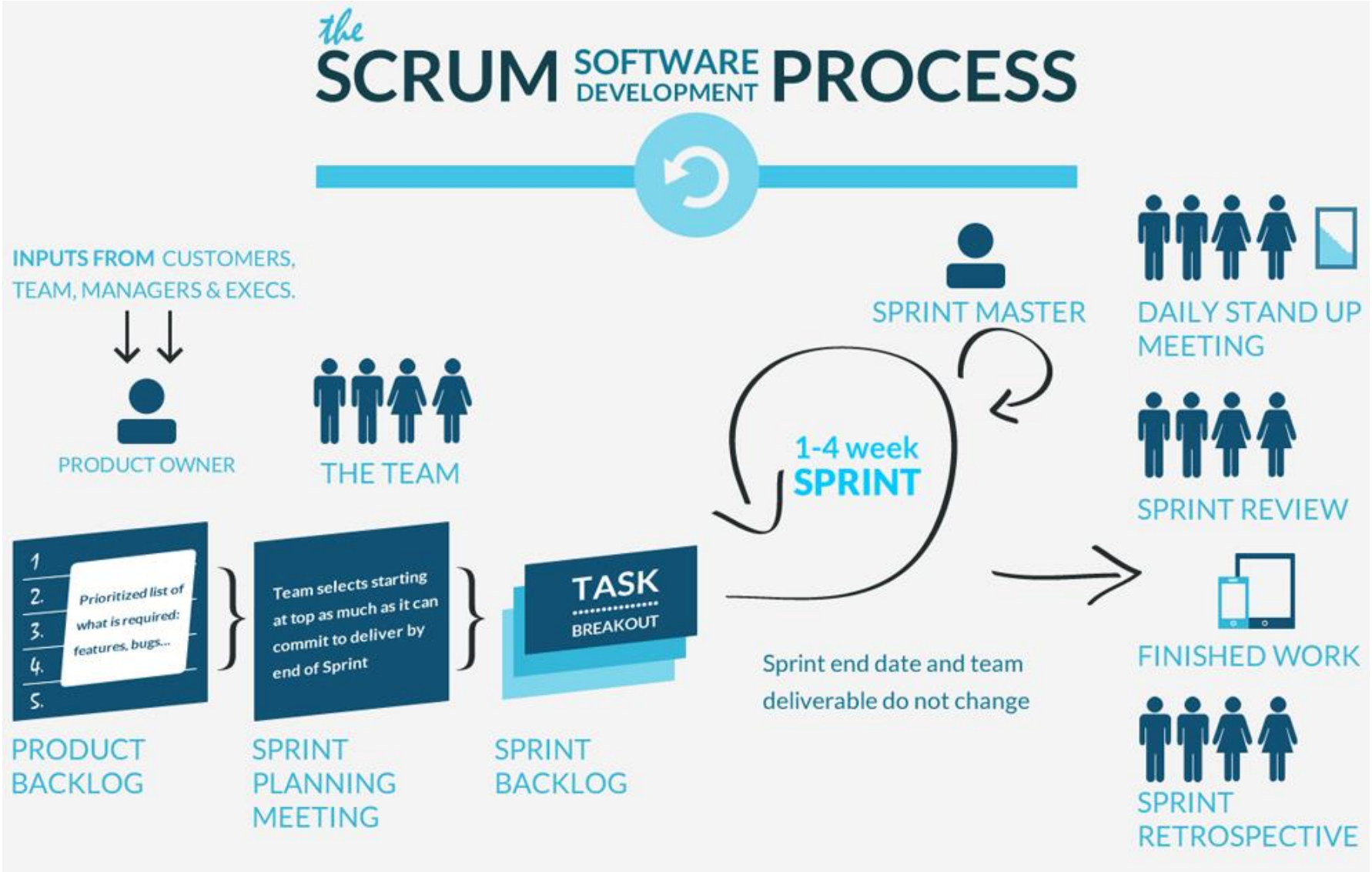
- ✓ The high amount of risk analysis, hence avoidance of Risk is enhanced.
- ✓ Good for large and mission-critical projects.
- ✓ Strong approval and documentation control.
- ✓ Additional Functionality can be added at a later date.
- ✓ Software is produced early in the software life cycle.

Disadvantages of Spiral Model

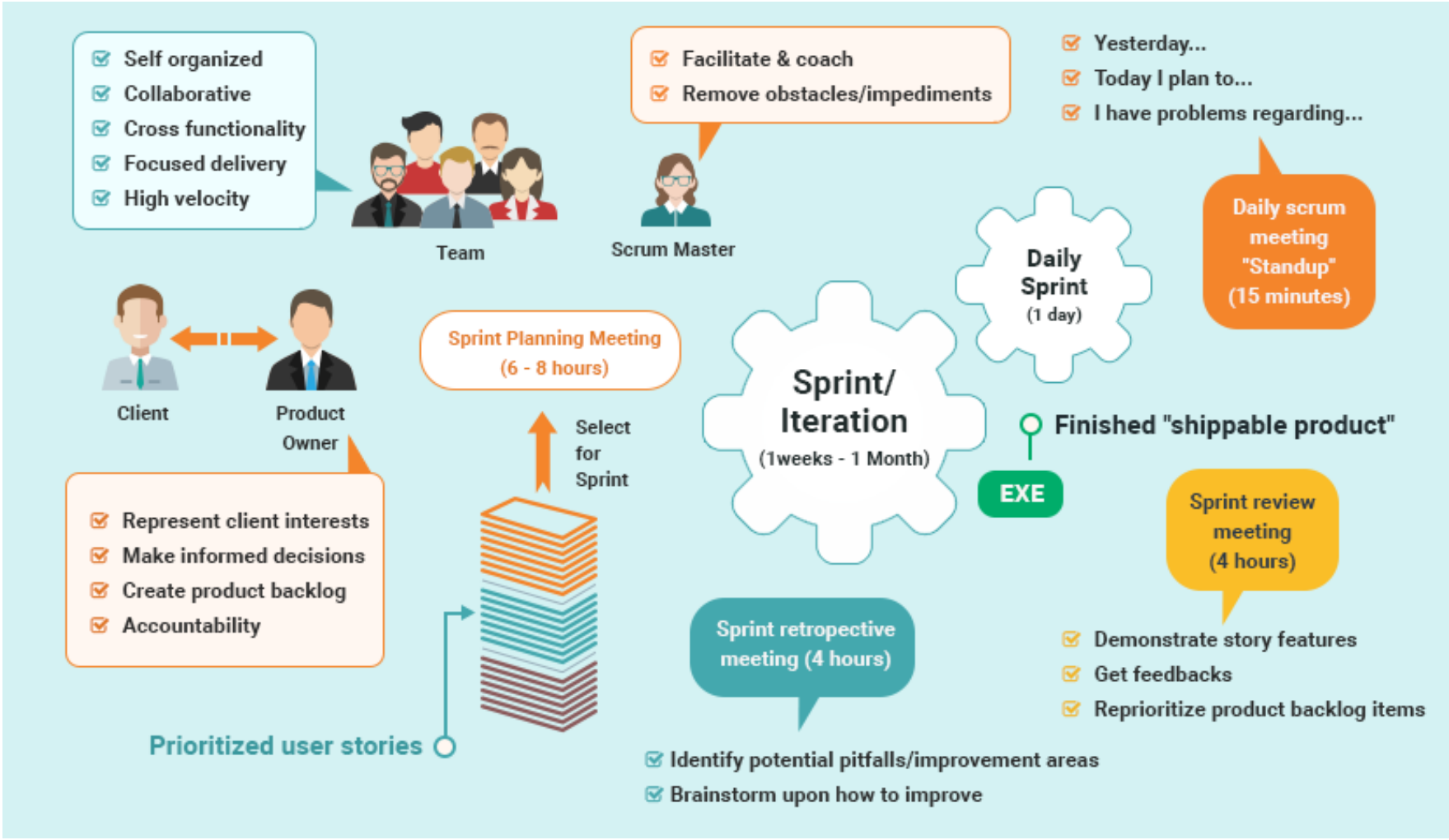
- ✓ Can be a costly model to use.
- ✓ Risk analysis requires highly specific expertise.
- ✓ Project's success is highly dependent on the risk analysis phase.
- ✓ Doesn't work well for smaller projects.

Agile

Modern Approach



SCRUM


Self organized
Collaborative
Cross functionality
Focused delivery
High velocity
Team
Facilitate & coach
Remove obstacles/impediments
Scrum Master
Yesterday...
Today I plan to...
I have problems regarding...
Daily scrum meeting "Standup" (15 minutes)
Daily Sprint (1 day)
Client
Product Owner
Sprint Planning Meeting (6 - 8 hours)
Sprint/ Iteration (1weeks - 1 Month)
Finished "shippable product"
EXE
Select for Sprint
Represent client interests
Make informed decisions
Create product backlog
Accountability
Sprint review meeting (4 hours)
Demonstrate story features
Get feedbacks
Reprioritize product backlog items
Sprint retropective meeting (4 hours)
Prioritized user stories
Identify potential pitfalls/improvement areas
Brainstorm upon how to improve