ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG VIỆT - HÀN
Vietnam - Korea University of Information and Communication Technology

SYSTEMS ANALYSIS AND DESIGN

Nguyen Thanh Binh, Nguyen Quang Vu, Le Viet Truong, Nguyen Thi Hanh, Vo Van Luong, Le Thi Bich Tra

Faculty of Computer Science

Vietnam - Korea University of Information and Communication Technology (VKU)

http://vku.udn.vn/

UML and Software Development Process

- Software Development Activities
- Object-Oriented Analysis and Design
- Software Development Processes
- UML and Software Development Processes

Main Software Development Activities

Requirements Gathering

Define requirement specification

Analysis

Define the conceptual model

Design

Design the solution / software plan

Implementation

Code the system based on the design

Integration and Test

Prove that the system meets the requirements

Deployment

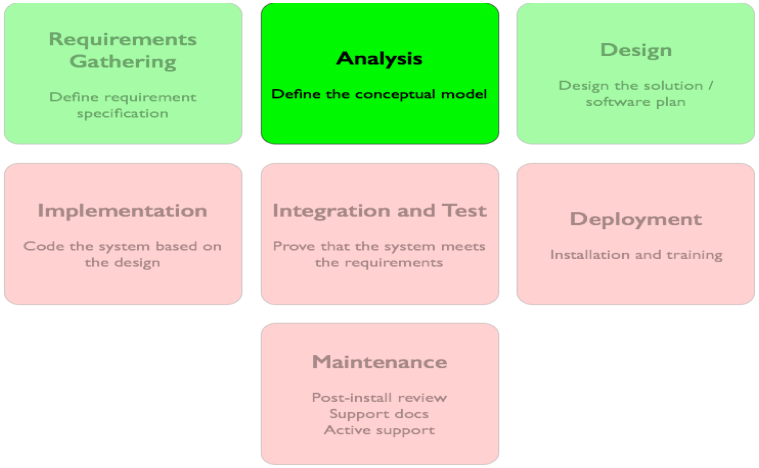
Installation and training

Maintenance

Post-install review
Support docs
Active support

Analysis emphasizes an investigation of the problem and requirements, rather than a solution.

During **object-oriented analysis**, there is an emphasis on finding and describing object or concepts in the problem domain.



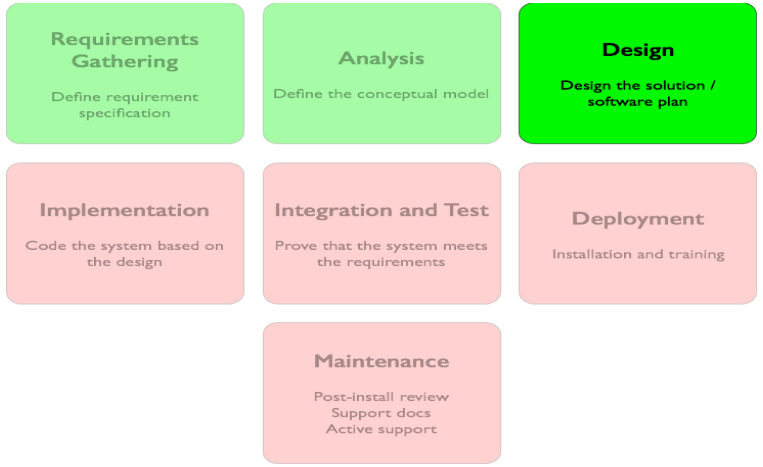
If a **cash register system** at the supermarket is desired

How will it be used?

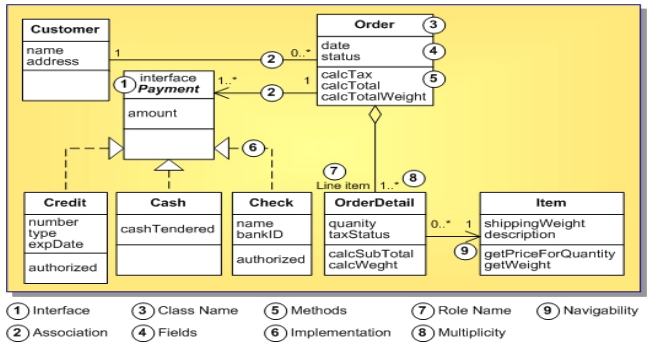
What are its functions?

Design emphasizes a conceptual solution in software that fulfils the requirements and "guides" the implementation.

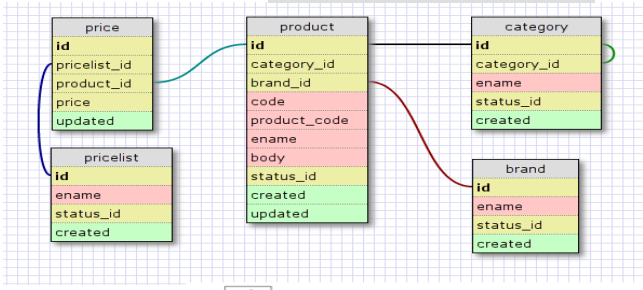
During **object-oriented design**, there is an emphasis on defining software objects and how they collaborate to fulfil the requirements.



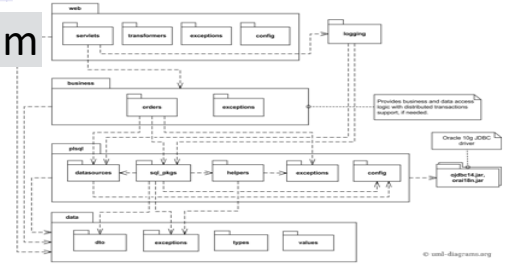
Class diagram



Database schema



Package diagram



Design specification of the **cash register system**

domain concept

representation in an object-oriented programming language

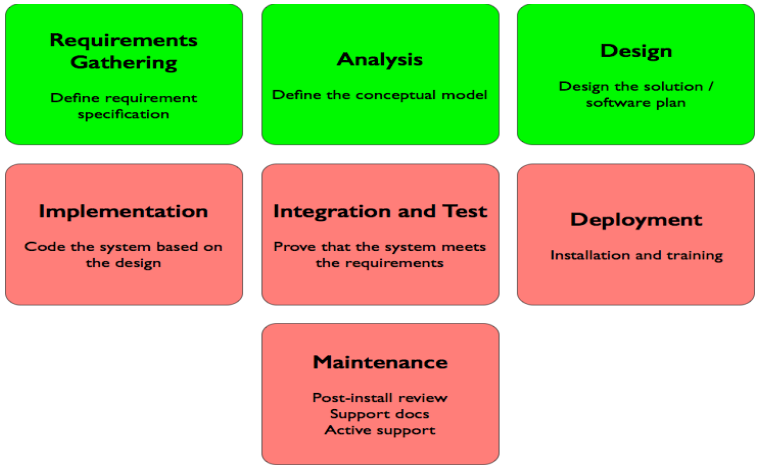
Plane
tailNumber

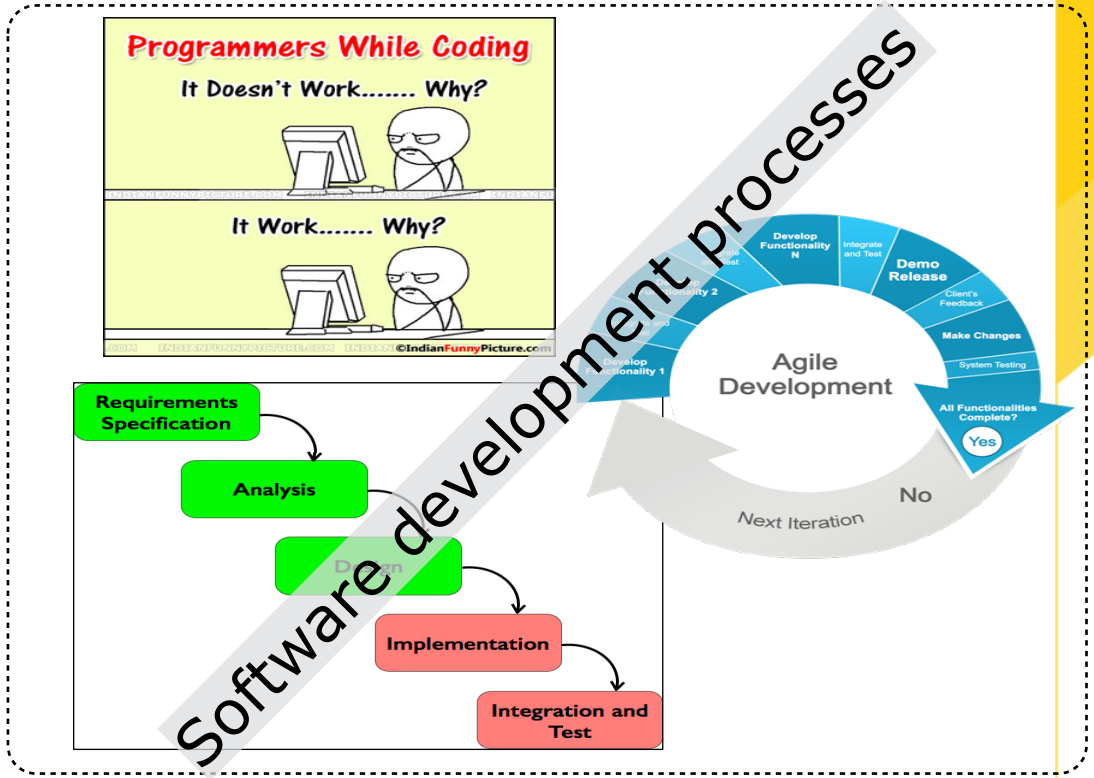
visualization of domain concept

```
public class Plane
{
private String tailNumber;

public List getFlightHistory() {...}
}
```

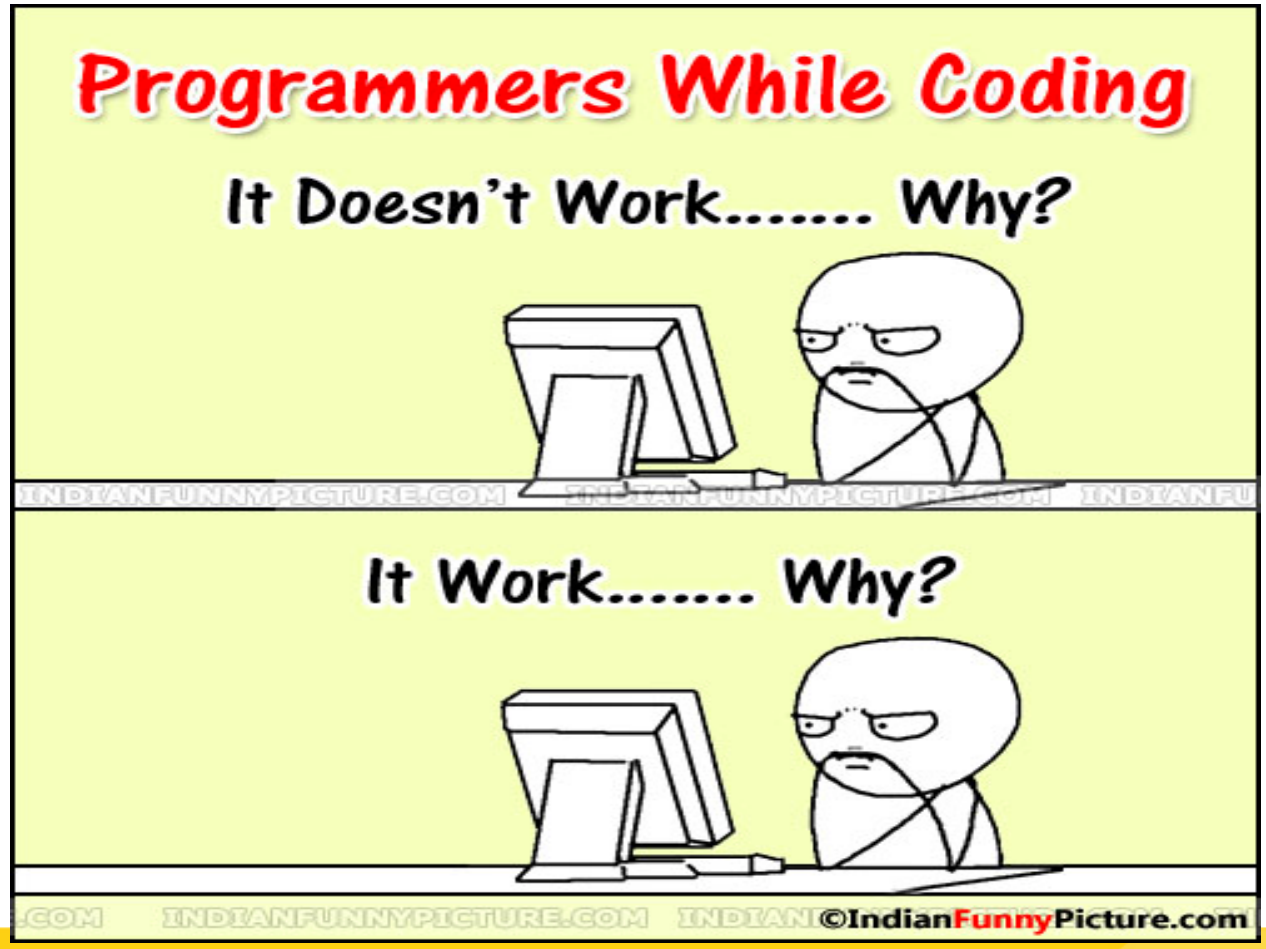
Software development process is a series of software development activities that a software program goes through when developed



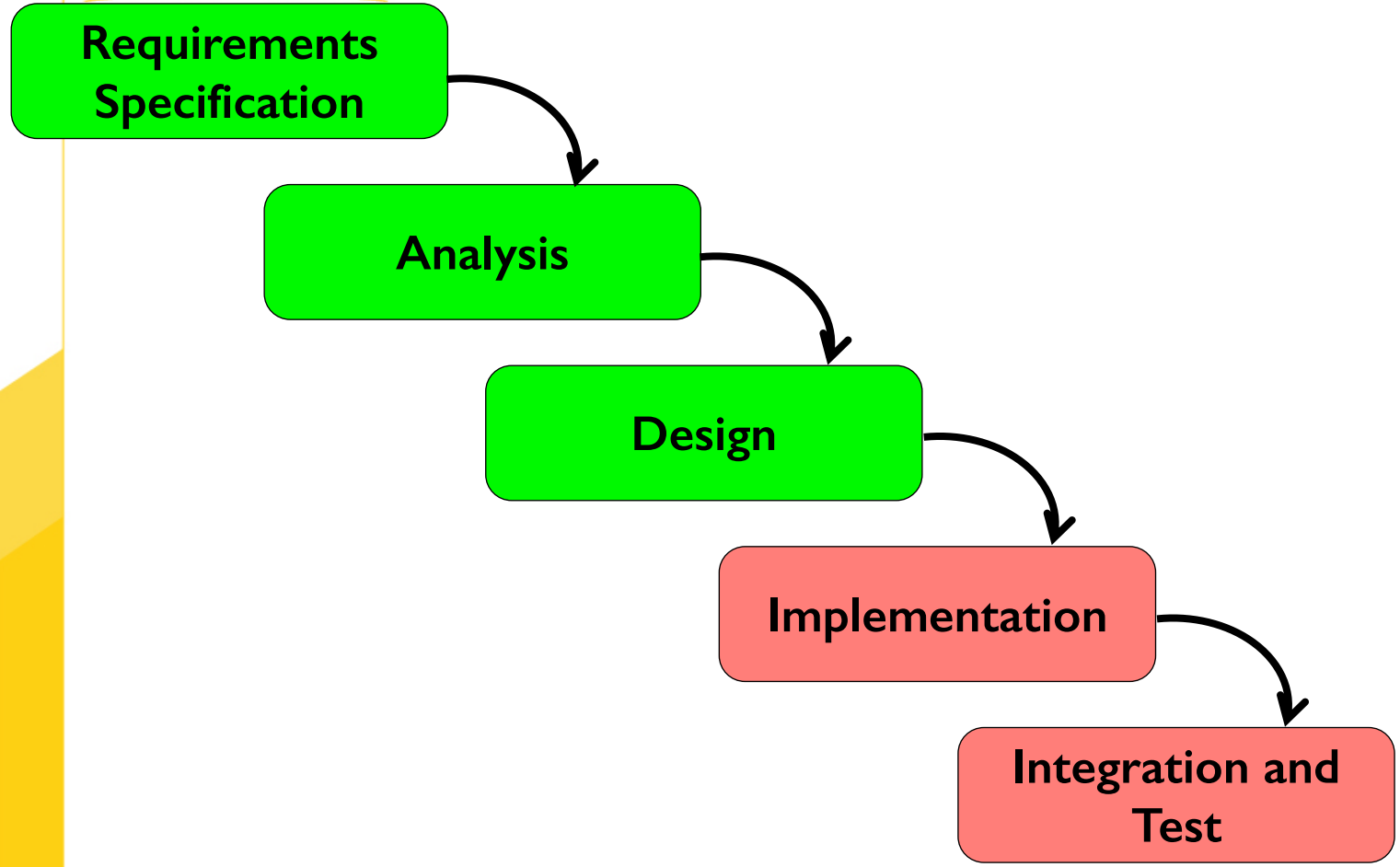


Ad-hoc Coding “process”

- Does not scale to large size project
- Does not scale to large development teams



Waterfall process

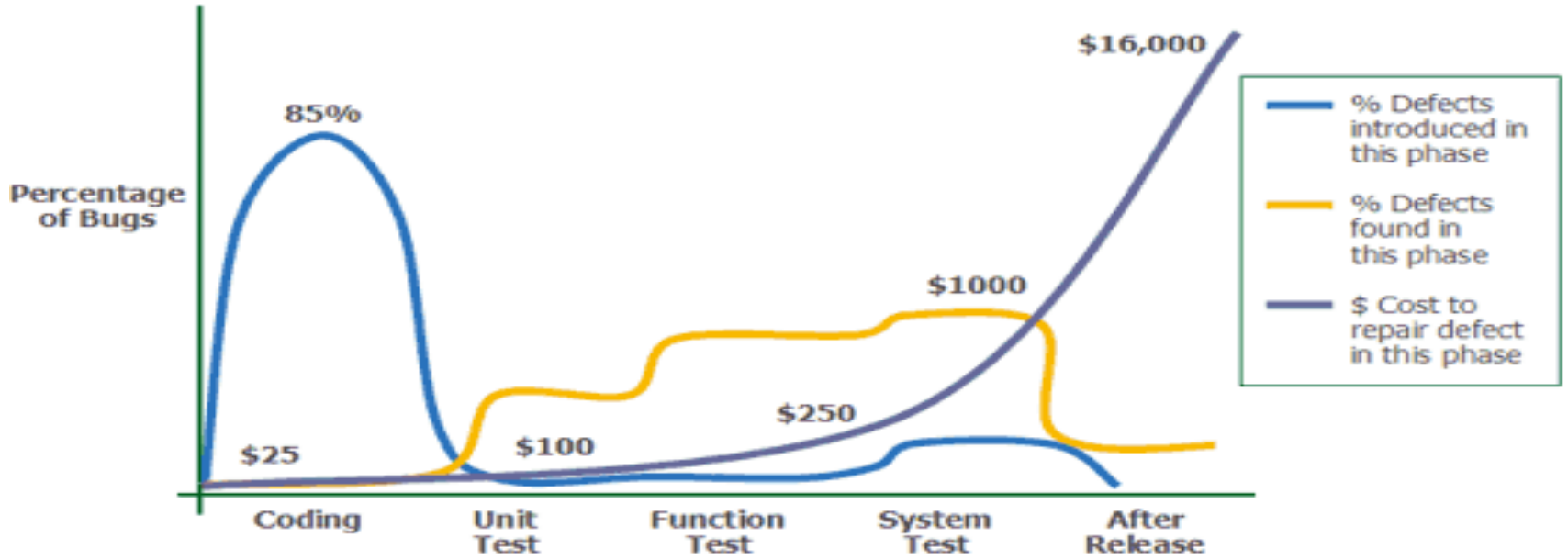


Sequential

- An phase is begun only when the previous has finished
- No return to previous phase

Critique of Waterfall process

- Responds poorly to changes and problems
- Substantial upfront document
- Assumes fixed specification - may not be what customer wants
- Fixes come very late - costlier to fix later time



Source: Applied Software Measurement, Capers Jones, 1996.

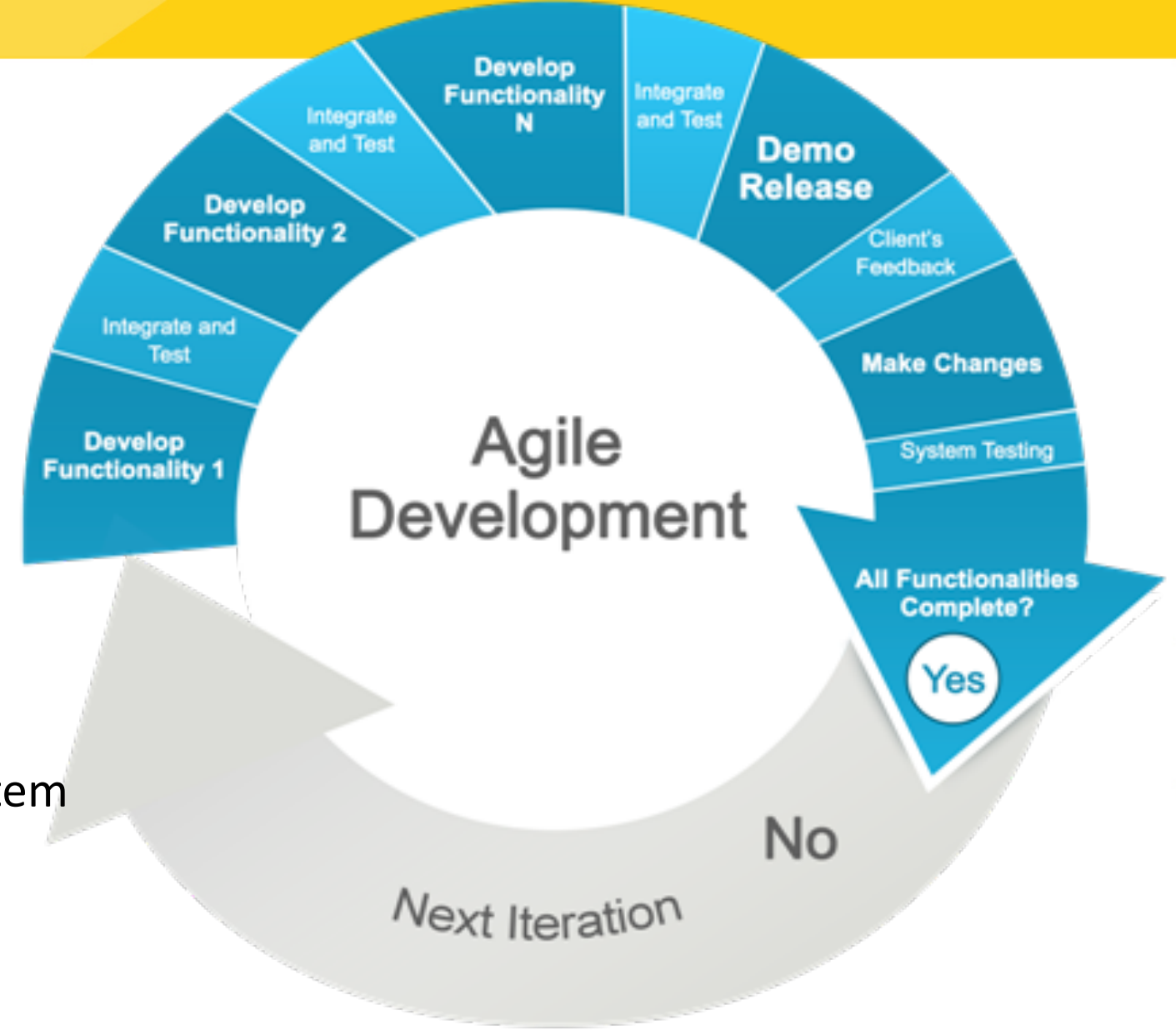
Iterative and Agile Development Processes

Facts of life

- Requirements change, changes break existing design.
- Coding up a design suggests flaws in design
- Testing identifies flaws in code - which could be design flaws
- Maintenance requires not only fixes but new features

Philosophy

- Embrace change
- Don’t do too much, too soon
- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan



Early coding, early testing of partial system in repeating cycles.
Development begins before all requirements are defined in detail.
Feedback is used to clarify evolving specification.

Benefits

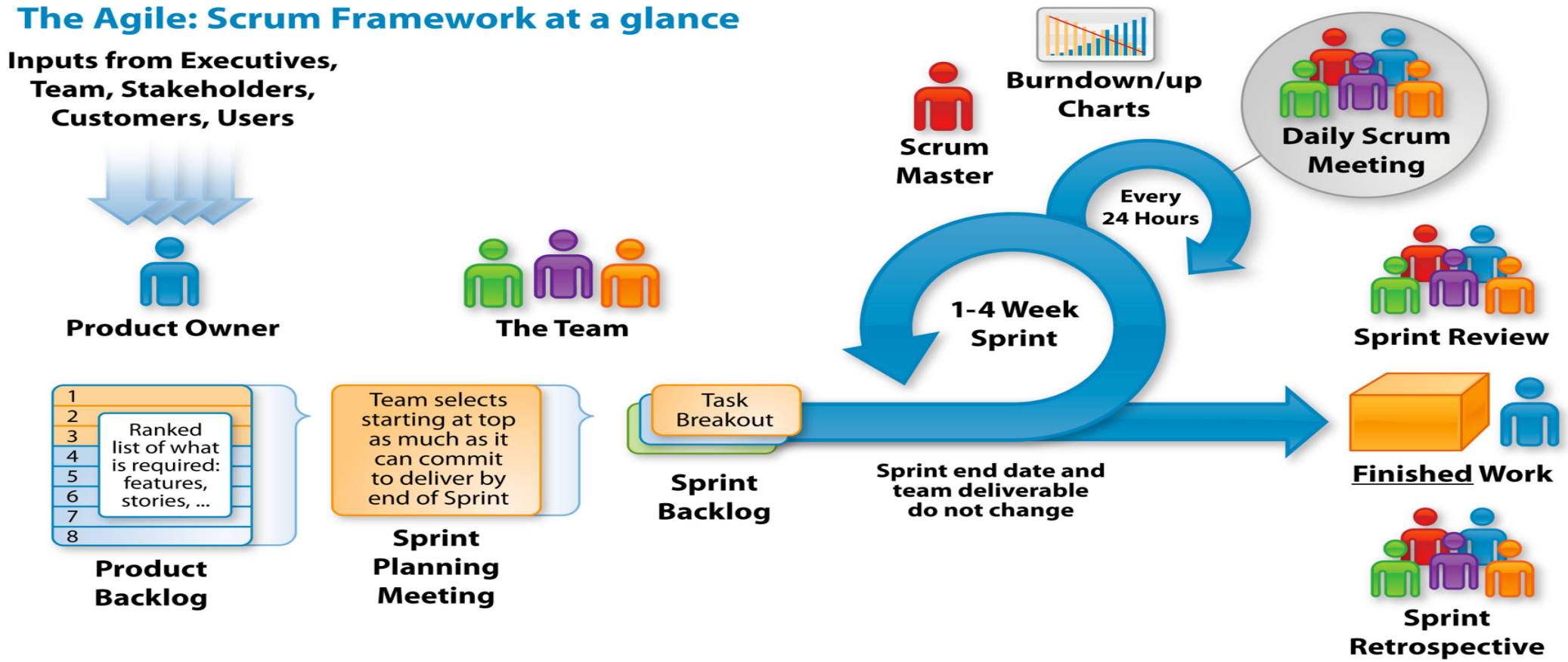
- Early rather than late mitigation of high risks
- Early visible progress
- Managed complexity - the team is not overwhelmed by “analysis paralysis” or very long and complex steps
- Early feedback, user engagement, and adaptation, leading to a redefined system that more closely meets the real needs of the stakeholders
- Feedback can also improve development process itself

Less project failure, better productivity, and lower defect rates

Agile software development methods

- Adaptive software development (ASD)
- Agile modeling
- Agile Unified Process (AUP)
- Crystal Clear Methods
- Disciplined agile delivery
- Dynamic Systems development method (DSDM)
- Extreme programming
- Feature-driven development (FDD)
- Lean software development
- Kanban
- Scrum

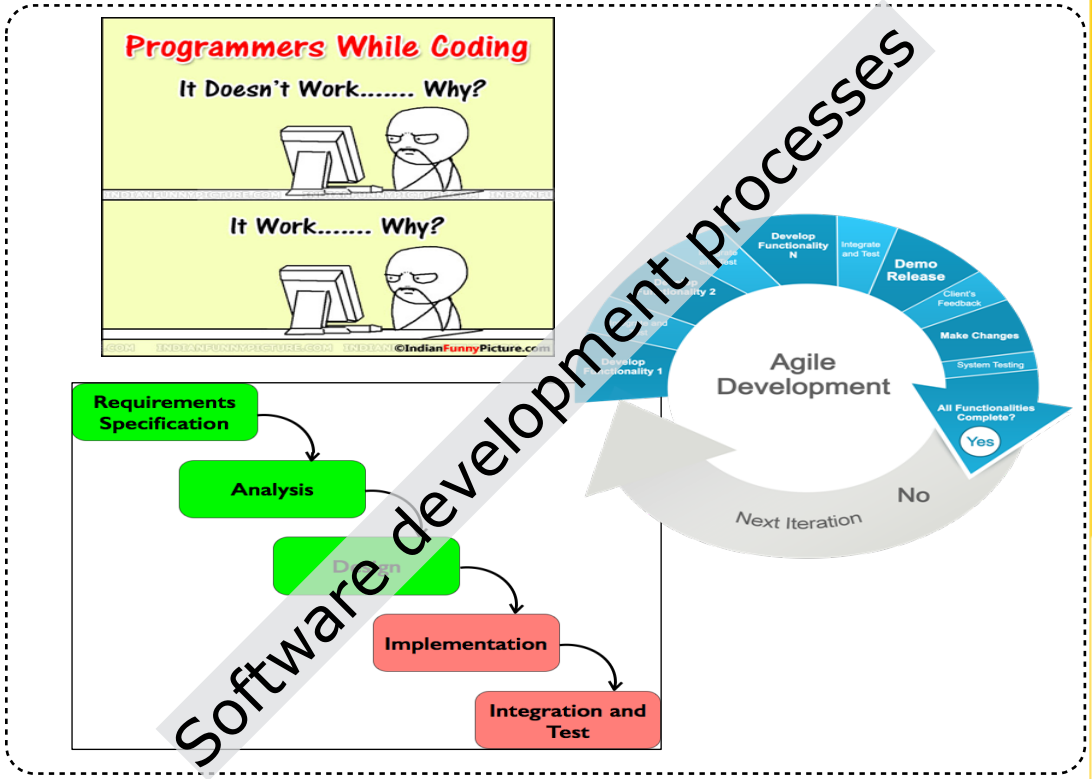
Scrum


The Agile: Scrum Framework at a glance
Inputs from Executives, Team, Stakeholders, Customers, Users
Product Owner
The Team
Scrum Master
Burndown/up Charts
Daily Scrum Meeting
Every 24 Hours
1-4 Week Sprint
Sprint Review
1
2
3
4
5
6
7
8
Ranked list of what is required: features, stories, ...
Product Backlog
Team selects starting at top as much as it can commit to deliver by end of Sprint
Sprint Planning Meeting
Task Breakout
Sprint Backlog
Sprint end date and team deliverable do not change
Finished Work
Sprint Retrospective

UML can be used in many software development process

UNIFIED MODELING LANGUAGE™

+



UML diagrams can be applied to several activities

	Requirements	Analysis	Design
Use-case	•		
Class, object		•	•
Activity		•	o
State		•	•
Interaction		o	•
Component			•
Deployment			•

o : possible usage
• : recommended usage